AUGUST 1986 VOL. 4, No. 8 $3.95

FOR IBM PERSONAL COMPUTER USERS

PC TECH JOURNAL®

OUT FROM THE SHADOW OF IBM

Compatibility and Performance on 286 Machines

COMPAQ DESKPRO 286

NINE-TRACK TAPE SYSTEMS

ANIMATION WITH SPRITES

Turbo GameWorks™

Also recently released, Turbo GameWorks is what you think it is: "Games" and "Works." Games you can play right away (like Chess, Bridge and Go-Moku), plus the Works—which is how computer games work. All the secrets and strategies of game theory are there for you to learn. You can play the games "as is" or modify them any which way you want. Source code is included to let you do that, and whether you want to write your own games or simply play the off-the-shelf games, Turbo GameWorks will give hours of diversion, education, and intrigue. George Koltanowski, Dean of American Chess, and former President, United States Chess Federation, reacted to Turbo GameWorks like this, "*With Turbo GameWorks, you're on your way to becoming a master chess player,*" and Kit Woolsey, writer, author, and twice Champion of the Blue Ribbon Pairs, wrote, "*Now play the world's most popular card game—Bridge . . . even program your own bidding or scoring conventions.*" Suggested retail: $69.95.

Turbo Database Toolbox™

A perfect complement to Turbo Pascal, because it contains a complete library of Pascal procedures that allows you to search and sort data and build powerful database applications. Having Turbo Database Toolbox means you don't have to reinvent the wheel each time you write a Turbo Pascal program. It comes with source code for a free sample database—right on disk. The database can be searched by keywords or numbers. Update, add, or delete records as needed. Just compile it and it's ready to go to work for you. Suggested retail: $69.95.

Turbo Graphix Toolbox™

It includes a library of graphics routines for Turbo Pascal programs. Lets even beginning programmers create high-resolution graphics with an IBM, Hercules,™ or compatible graphics adapter. Our Turbo Graphix Toolbox includes all the tools you'll ever need for complex business graphics, easy windowing, and storing screen images to memory. It comes complete with source code, ready to compile. Suggested retail: $69.95.

Technical Specifications:

TURBO PASCAL 3.0 Minimum memory:128K; includes 8087 and BCD features for 16-bit MS-DOS and CP/M-86 systems. CP/M-80 version minimum memory: 48K; 8087 and BCD features not available

TURBO DATABASE TOOLBOX Minimum memory: 128K. CP/M-80 minimum memory: 48K. Requires Turbo Pascal 2.0 or later.

TURBO GRAPHIX TOOLBOX* Minimum memory: 192K. Requires PC/MS-DOS 2.0 or later, Turbo Pascal 3.0, and IBM CGA, Hercules Monochrome Card or equivalent.

TURBO TUTOR 2.0 Minimum memory: 192K. CP/M-80 version minimum memory 48K. Requires PC/MS-DOS 2.0 or later and Turbo Pascal 3.0.

TURBO EDITOR TOOLBOX* Minimum memory: 192K. Requires PC/MS-DOS 2.0 or later and Turbo Pascal 3.0.

TURBO GAMEWORKS* Minimum memory: 192K. Requires PC/MS-DOS 2.0 or later and Turbo Pascal 3.0.

TURBO PROLOG* Minimum memory: 384K.

REFLEX: THE ANALYST* Minimum memory: 384K. Requires IBM CGA, Hercules Monochrome Card or equivalent. Works with Intel's AboveBoard-PC and -AT; AST's RAMpage! and RAMpage! AT; Quadram's Liberty-PC and -AT; Tecmar's 640 Plus; IBM's EGA and 3270/PC; AT&T's 6300 and many others.

REFLEX WORKSHOP* Minimum memory: 384K. Requires Reflex: The Analyst. Two disk drives or hard disk recommended.

TURBO LIGHTNING* Minimum memory: 256K. Two disk drives required. Hard disk recommended.

LIGHTNING WORD WIZARD* Minimum memory: 256K. Requires Turbo Lightning. Turbo Pascal 3.0 required to edit source code.

SIDEKICK* Minimum memory: 128K.

TRAVELING SIDEKICK* Minimum memory: 256K.

SUPERKEY* Minimum memory: 128K.

*For IBM PC, AT, XT, PCjr and true compatibles only, running PC/MS-DOS 2.0 or later.

Recognition for Borland International has come from business, trade, and media, and includes both product awards and awards for technical excellence and marketing. Borland was named "Company of the Year" by PC Magazine; Sidekick, the #1 best seller for the IBM PC, was named "Product of the Year" by InfoWorld; Turbo Pascal was selected one of PC Week's Top 10 Products for 1984; SuperKey won one of PC Magazine's "Best of 1985" awards; Reflex, The Analyst was recognized in the "Software Products of the Year" awards by InfoWorld; and Reflex and SideKick were both nominated for British Micro Awards in 1986.

News from Borland International! Vol. 1 No. 1

We frequently surprise people with inventive, imaginative software, and people frequently surprise us with the way *they* use it.

For example, you'll read on this page how Michael J. Watkins of the Petroleum Technology Center in Houston, Texas, used Turbo Pascal® (and Turbo Graphix Toolbox™ and Turbo Tutor®) to cut down the tedium and time in creating Circular Performance Profile Charts (CPPCs).

We didn't know they existed, but you learn something new every day!

Applications like CPPCs might not fit your exact needs, but at the same time they might stimulate fresh ideas in your mind about how you can put Turbo Pascal and the Turbo Pascal family to work for you.

And thank you for your interest in and support for Borland International.

Philippe Kahn,
President, Borland International

INSIDE STORIES!

- Turbo Pascal 3.0, already described by *PC Magazine* as "*Language deal of the century*," is now an even better deal than that, because we've included the most popular options (BCD reals and 8087 support). What used to cost $124.95 is now only $99.95!
- Completely new Turbo Tutor 2.0 now available. New software. New manual. New split screens. New quizzes. Only $39.95. Upgrades available under Borland's "Almost-Free" upgrade plan. Details inside.

LATE NEWS!

- June/July Special Artificial Intelligence Issue of The *Micro Technical Journal* says, "*Turbo Prolog looks like it's going to be a winner, for both the beginner and professional programmer.*"

Turbo Pascal deliberately programmed to go around in circles

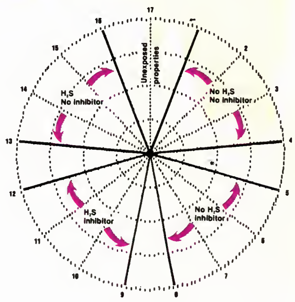
Circular charts (or CPPCs) are used by Michael J. Watkins of the Petroleum Technology Center in Houston, Texas, to plot a single performance property for a large number of elastomers, which have elastic, rubber-like properties.

Mr. Watkins wrote us saying, "*Because CPPCs condense a lot of data in one graphic, they can be very tedious and time-consuming to draw.*"

What he did to solve those problems was to write a Turbo Pascal program for IBM® personal computers to "*generate these charts quickly and easily.*"

He used Turbo Pascal "*because it has a companion set of very powerful graphics programs (Turbo Graphix Toolbox) which greatly simplifies the required programming.*

Turbo Pascal is not a difficult language to use and can be easily learned by persons who can program in FORTRAN or BASIC. An excellent tutorial (Turbo Tutor) is available for the novice or experienced programmer. The Turbo Pascal products are also very moderately priced."



"The computer is no better than its program."
Elting Morison, author of "Men, Machines and Modern Times"

COMPLETELY NEW VERSION!

Turbo Tutor® 2.0

Just released (July '86), the new Turbo Tutor can take you from "What's a computer?" on through to complex data structures, assembly languages, trees, tips on writing long programs in Turbo Pascal, and a high level of expertise. Source code for everything is included. New split screens allow you to put source text in the bottom half of the screen and run the examples in the top half. There are quizzes that ask you, show you, tell you, teach you. You get a 450-page manual—which is not as daunting as it sounds, because unlike many software manuals, it was not written by orangutans. (With our all "almost-free" upgrade, you can upgrade to Turbo Tutor 2.0 by sending us your master diskettes, proof of purchase, and $10.00, which covers shipping and handling.) Suggested retail: $39.95.

NEW SPECIAL!

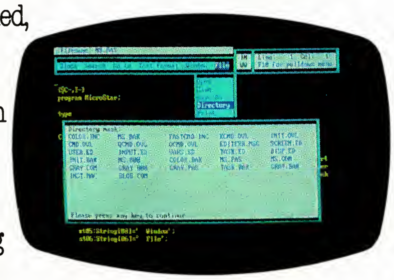
Turbo Pascal® 3.0

"*For the IBM PC, the benchmark Pascal compiler is undoubtedly Borland International's Turbo Pascal,*" says Gary Ray of *PC Week*. We and more than 500,000 other people around the world think Mr. Ray got that right.

Since launch, Turbo Pascal has become the *de facto* worldwide standard in high-speed Pascal compilers. Described by Jeff Duntemann of *PC Magazine* as the "*Language deal of the century,*" Turbo Pascal is now an even better deal than that—because we've *included* the most popular options (BCD reals and 8087 support). What used to cost $124.95 is now only $99.95! You now get a lot more for a lot less: the compiler, a completely integrated programming environment, and BCD reals and 8087 support—all for a suggested retail of only $99.95.

Turbo Editor Toolbox™

Recently released, we called our new Turbo Editor Toolbox a "construction set to write your own word processor." Peter Feldmann of *PC Magazine* covered it pretty well with, "*A 'write your own word processor' program for intermediate level programmers, with lots of help in the form of prewritten procedures covering everything from word wrap to pull-down windows.*" Source code is included, and we also include MicroStar, a full-blown text editor with pull-down menus and windowing. It interfaces directly with Turbo Lightning to let you spell-check your MicroStar files. Jerry Pournelle of *BYTE* magazine said, "*The new Turbo Editor Toolbox is the Turbo Pascal source code to just about anything you ever wanted a PC-compatible text editor to do.*" Suggested retail: $69.95.



Borland's Business Productivity Programs:

Reflex: The Analyst™ Analytical database manager. Provides complete new look at data normally hidden by programs like 1-2-3® and dBASE.® Best report generator for 1-2-3.

NEW! **Reflex Workshop™** Important new addition to Reflex: The Analyst. Gives you 22 different templates to run your business right.

SideKick® Complete RAM-resident desktop management includes notepad, dialer, calculator and more.

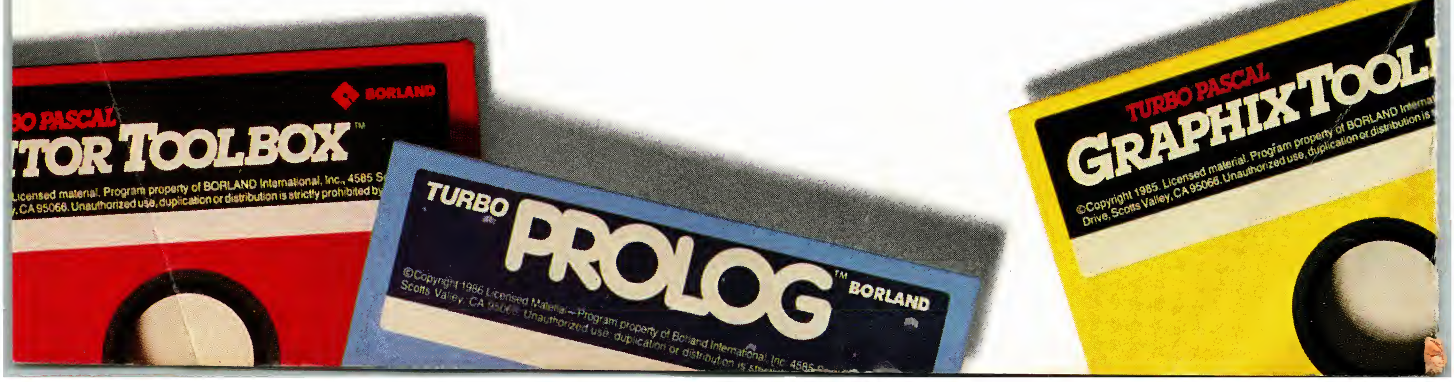
Traveling SideKick™ Electronic version of business/personal diaries, daytime organizers; works with your SideKick files; important professional tool.

SuperKey® Keyboard enhancer. Simple macros turn 1000 keystrokes into 1. Also encrypts your files to keep confidential files confidential.

Borland's Electronic Reference Programs:

Turbo Lightning™ Works with all your programs and checks your spelling *while you type*! Includes 80,000-word Random House® Concise Dictionary and 50,000-word Random House Thesaurus. Forerunner of Turbo Lightning Library.

Lightning Word Wizard™ Includes ingenious crossword solver and six other word challenges. If you're into programming, Lightning Word Wizard is also a development toolbox and the technical reference manual for Turbo Lightning.



Step-by-step tutorial, demo programs with source code included!

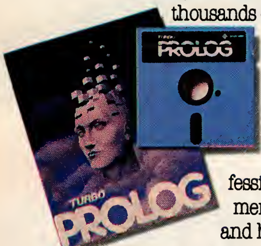
Borland's new Turbo Prolog is the powerful, completely natural introduction to Artificial Intelligence

Prolog is probably one of the most powerful computer programming languages ever conceived, which is why we've made it our *second* language—and "turbo-charged" it to create Turbo Prolog.™

Our new Turbo Prolog, the natural language of Artificial Intelligence, brings supercomputer power to your IBM® PC and introduces you step-by-step to the fascinating new world of Artificial Intelligence. And does all this for an astounding $99.95.

Turbo Prolog is to Prolog what Turbo Pascal® is to Pascal!

Our Turbo Pascal astonished everyone who thought of Pascal as "just another language." We changed all that—and now Turbo Pascal is the de facto worldwide standard, with hundreds of thousands of enthusiasts and users in universities, research centers, schools, and with professional programmers, students, and hobbyists.



You can expect at least the same impact from Turbo Prolog, because while Turbo Prolog is the most revolutionary and natural programming language, it is also a complete development environment—just like Turbo Pascal.

Minimum memory: 384K

> "Turbo Prolog offers generally the fastest and most approachable implementation of Prolog."
>
> *Darryl Rubin, AI Expert*

Even if you've never programmed before, our free tutorial will get you started right away

You'll get started right away because we have included a complete step-by-step tutorial as part of the 200-page Turbo Prolog Reference Manual. Our tutorial will take you by the hand and teach you everything you're likely to need to know about Turbo Prolog and artificial intelligence.

For example: once you've completed the tutorial, you'll be able to design your own expert systems utilizing Turbo Prolog's powerful problem-solving capabilities.

Think of Turbo Prolog as a high-speed electronic detective. First you feed it information and teach it rules. Then Turbo Prolog "thinks" the problem through and comes up with all the reasonable answers—almost instantly.

If you think that this is amazing, you just need to remember that Turbo Prolog is a 5th-generation language—and the kind of language that 21st century computers will use routinely. In fact, you can compare Turbo Prolog to Turbo Pascal the way you could compare Turbo Pascal to machine language.

You get the complete Turbo Prolog programming system for only $99.95

You get a complete Turbo Prolog development system including:

- The lightning-fast Turbo Prolog incremental compiler and the interactive Turbo Prolog editor.
- The 200-page reference manual which includes the step-by-step Turbo Prolog tutorial.
- The *free* GeoBase™ natural query language database including commented source code on disk—ready to compile. GeoBase is a complete database designed and developed around U.S. geography. It includes cities, mountains, rivers, and highways, and comes complete with natural query language. Use GeoBase immediately "as is," or modify it to fit your own interests.

So don't delay—don't waste a second—get Turbo Prolog now. $99.95 is an amazingly small price to pay to become an immediate authority—an instant expert on artificial intelligence! The 21st century is only one phone call away.

BORLAND
INTERNATIONAL
4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066
(408) 438-8400 TELEX: 172373

Vive la différence

CIRCLE NO. 254 ON READER SERVICE CARD

YES! I want the best!

For credit card orders or the dealer nearest you
call (800) 255-8008
in CA call (800) 742-1133

Copies	Product	Price	Totals
___	Turbo Prolog	$99.95	$ ______
___	Reflex: The Analyst	149.95*	$ ______
___	Reflex Workshop	69.95*	$ ______
___	Reflex, Reflex Workshop	199.95*	$ ______
___	Turbo Pascal 3.0 w/8087 & BCD	99.95	$ ______
___	Turbo Pascal for CP/M-80	69.95	$ ______
___	Turbo Database Toolbox	69.95	$ ______
___	Turbo Graphix Toolbox	69.95	$ ______
___	Turbo Tutor 2.0	39.95	$ ______
___	Turbo Editor Toolbox	69.95	$ ______
___	Turbo GameWorks	69.95	$ ______
___	Turbo Lightning	99.95	$ ______
___	Lightning Word Wizard	69.95	$ ______
___	Turbo Lightning, Lightning Word Wizard	149.95	$ ______
___	SideKick	84.95	$ ______
___	Traveling SideKick	69.95*	$ ______
___	SideKick, Traveling SideKick	125.00*	$ ______
___	SuperKey	69.95	$ ______
	Outside USA add $10 per copy CA and MA res. add sales tax		$ ______
	Amount enclosed		$ ______

Prices include shipping to all US cities.

Carefully describe your computer system:

Mine is: ___ 8-bit ___ 16-bit

I use: ___ PC-DOS ___ CP/M-80 ___ MS-DOS ___ CP/M-86

My computer's name and model is: ______________________

The disk size I use is: ☐ 3½" ☐ 5¼" ☐ 8"

Payment: VISA MC Money order Check

Credit card expiration date ___/___

Card # ______________________

NOT COPY PROTECTED
****60-DAY MONEY-BACK GUARANTEE**

Name: ______________________

Shipping Address: ______________________

City: ______________________

State: ____________ Zip: ____________

Telephone: ______________________

CODs and purchase orders WILL NOT be accepted by Borland. Outside USA make payment by bank draft, payable in US dollars drawn on a US bank.

*Limited Time Offer until September 1, 1986.

**YES, if within 60 days of purchase you find that this product does not perform in accordance with our claims, call our customer service department and we will gladly arrange a refund.

Prices subject to change without notice.

Turbo Prolog 1.0 Technical Specifications

Compiler: Incremental compiler generating native in-line code and linkable object modules. The linking format includes a linker and is compatible with the PC-DOS linker. Large memory model support. Compiles over 2500 lines per minute on a standard IBM PC.

Interactive Editor: The system includes a powerful interactive text editor. If the compiler detects an error, the editor automatically positions the cursor appropriately in the source code. At run-time, Turbo Prolog programs can call the editor, and view the running program's source code.

Type System: A flexible object-oriented type system is supported.

Windowing Support: The system supports both graphic and text windows.

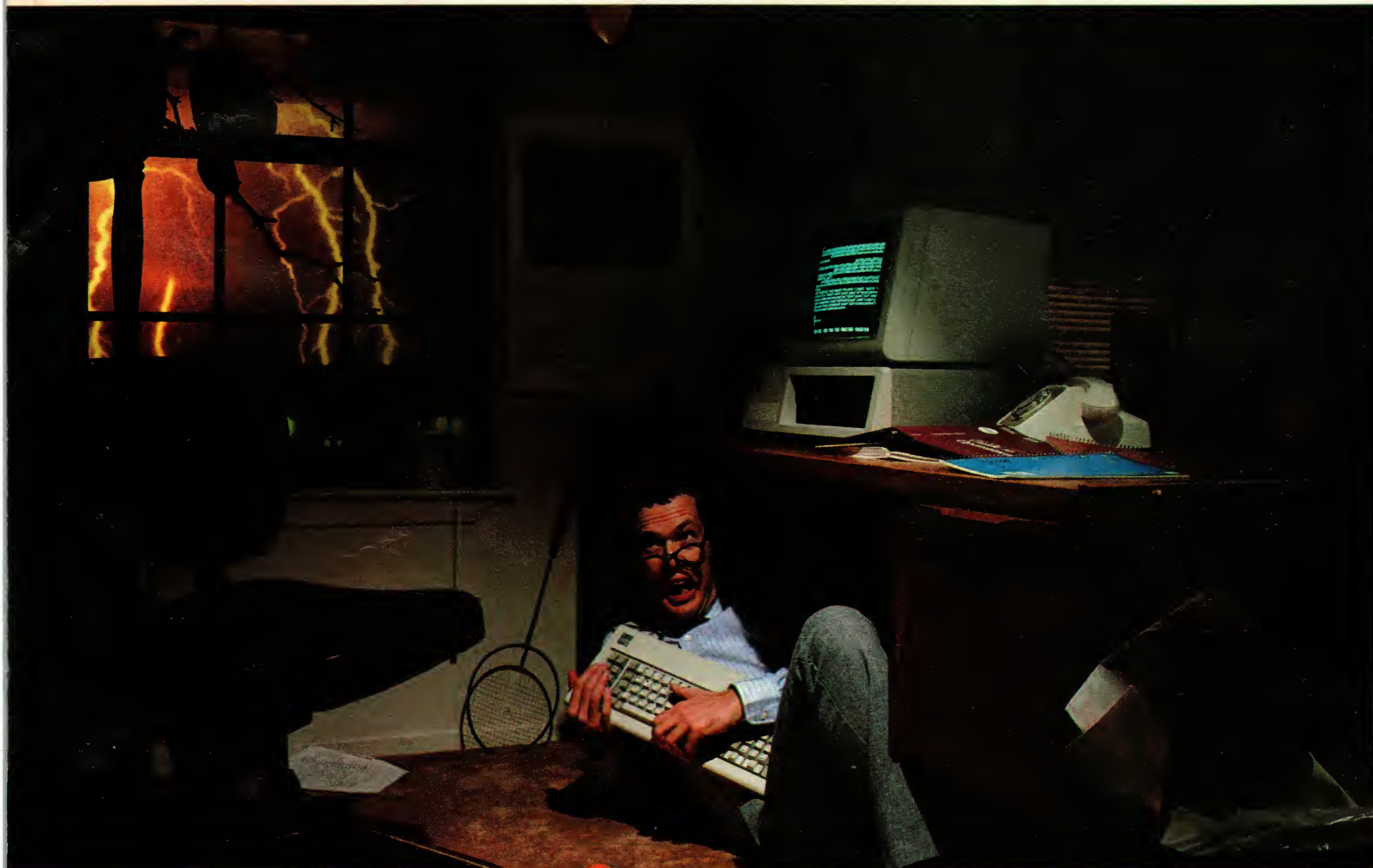
Input/Output: Full I/O facilities, including formatted I/O, streams, and random access files.

Numeric Ranges: Integers: −32767 to 32767; Reals: 1E−307 to 1E+308.

Debugging: Complete built-in trace debugging capabilities allowing single stepping of programs.

GP6

If lightning still scares you, you're using the wrong file manager.



Be sure. Btrieve.®

Lightning may strike. But it doesn't have to destroy your database.

Btrieve® file management offers automatic file recovery after a system crash. So accidents and power failures don't turn into database disasters. Your Btrieve-based applications will come up when the lights come back on.

Fast. Btrieve is lightning fast, too. Its b-tree file structure automatically balances—you never waste time reorganizing the index. And Btrieve is written in Assembly language especially for the IBM PC. The result: electrifying speed on file maintenance routines. Applications that run fast. Users who don't waste time waiting.

The standard for networking. When your application requires multi-user file sharing, Btrieve/N (network version) sets the standard for the industry's most popular LANs: IBM's PC Network, Netware, PC Net and EtherSeries. And now Btrieve sets the standard for multi-user systems: XENIX, Multi-Link Advanced and others. Btrieve/N offers **safe** network file management that coordinates simultaneous updates and protects against lost data.

Fully-relational data management. Using SoftCraft's entire family of products gives you a complete, fully-relational database management system. Rtrieve™ adds report writing capabilities for generating the reports you need. Xtrieve™ speeds users through database queries with interactive, on-screen menus—no command language or special syntax.

For professional programmers. Btrieve is the fast, reliable answer for all your application development. In any development language—BASIC, Pascal, COBOL, C, FORTRAN and APL. With multikey access to records. Unlimited records per file. Duplicate, modifiable, and segmented keys.

With Btrieve, you can develop better applications faster. And know they'll be safe if lightning strikes.

NO ROYALTIES

Suggested retail prices: Btrieve, $245; Btrieve/N, $595; Xtrieve, $245; Xtrieve/N, $595; Rtrieve, $145; Rtrieve/N, $345. Requires PC-DOS or MS-DOS 1.X, 2.X, or 3.X.

Btrieve is a registered trademark and Xtrieve and Rtrieve are trademarks of SoftCraft Inc.

SoftCraft Inc.
P.O. Box 9802 #917
Austin, Texas 78766
(512) 346-8380 Telex 358 200

CIRCLE NO. 201 ON READER SERVICE CARD

PC TECH JOURNAL

AUGUST 1986 VOLUME 4, NUMBER 8

Cover photograph · Marc David Cohen

PC TECH JOURNAL

VOL. 4, NO. 8

PUBLISHER: *Newton Barrett*
EDITOR: *Will Fastie*

EDITORIAL
MANAGING EDITOR: *Marjory Spraycar*
EXECUTIVE EDITOR: *Julie Anderson*
SENIOR TECHNICAL EDITOR: *Jim Shields*
TECHNICAL EDITORS: *Jeff Duntemann, Caroline Halliday*
ASSOCIATE TECHNICAL EDITOR: *Dan Beale*
SENIOR COPY EDITOR: *Susan Holly*
COPY EDITOR: *Gail Shaffer*
PROOFREADERS: *Kathleen Peddicord, Elizabeth Wardlaw*
NEW PRODUCTS EDITOR: *Carole Autenzio*
ADMINISTRATIVE ASSISTANT: *Diana Carey*
CONSULTING EDITORS: *Thomas V. Hoffmann, Richard M. Foard*
CONTRIBUTING EDITORS: *Steven Armbrust, Michael Covington, Ted Forgeron, Augie Hansen, Ted Mirecki, Max Stul Oppenheimer*

ART & PRODUCTION
ART DIRECTOR: *Ina Saltz*
ASSOCIATE ART DIRECTOR: *Sharon Reuter*
ASSISTANT ART DIRECTOR: *Monique Cubicciotti*
ART SECRETARY: *Sabrina Reynolds*
ADVERTISING PRODUCTION MANAGER: *Lisa Franey Ducey*
EDITORIAL PRODUCTION COORDINATOR: *James Olsson*

ADVERTISING SALES
ADVERTISING DIRECTOR: *Rita Burke*
MARKETING DIRECTOR: *Gayl Sorota*
MARKETING COORDINATOR: *Julie Henderson*
ADVERTISING COORDINATOR: *Michele Fischetti*
DISTRICT MANAGERS: *Rosemarie Caruso, Jan Schultz—East Coast; Pat Toohey—Mid-Atlantic; Arlene Braithwaite—Midwest; Ted Bahr, Bill Bush, Phyllis Egan—West Coast*
ACCOUNT REPRESENTATIVES: *Jane Anderson—East Coast; Pam Sigal, Jane Anderson—West Coast*

CIRCULATION
CIRCULATION MANAGER: *Charles Mast*
CIRCULATION SALES DEVELOPMENT: *Daniel Rosensweig*
MEDIA MANAGER: *Melinda Kendall*

ZIFF-DAVIS PUBLISHING COMPANY, *a division of Ziff Communications Company*
PRESIDENT: *Kenneth H. Koppel*
SENIOR VICE PRESIDENT, Marketing: *Paul Chook*
VICE PRESIDENT, Operations: *Baird Davis*
VICE PRESIDENT, Controller: *John Vlachos*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Circulation: *Alicia Marie Ivans*
VICE PRESIDENT, Circulation Services: *James Ramaley*
VICE PRESIDENT, Marketing Services: *Ann Pollak Adelman*
VICE PRESIDENT, Development: *Seth Alpert*
VICE PRESIDENT: *Hugh Tietjen*
BUSINESS MANAGER: *Gary A. Gustafson*
PRODUCTION DIRECTOR: *Walter J. Terlecki*

ZIFF COMMUNICATIONS COMPANY
PRESIDENT: *Philip B. Korsant;* EXECUTIVE VICE PRESIDENT: *James D. Dunning, Jr.;* SENIOR VICE PRESIDENTS: *Philip Sine, Kenneth H. Koppel;* VICE PRESIDENTS: *Laurence Usdin, William L. Phillips, J. Malcolm Morris, Steven C. Feinman;* TREASURER: *Selwyn Taubman;* SECRETARY: *Bertram A. Abrams*

EDITORIAL OFFICE
PC TECH JOURNAL, The World Trade Center, Suite 211, Baltimore, MD 21202. 301/576-0770. FAX (group 3): 301/576-9603. MCIMail: PCTECH. PCTECHline: 301/576-PCTJ (7285). Telex: 6502565932 MCI.

ADVERTISING OFFICES
(East Coast/Midwest) One Park Ave., New York, NY 10016. 212/503-5185. (West Coast) 3460 Wilshire Blvd., Los Angeles, CA 90010. 213/387-2100; 11 Davis Drive, Belmont, CA 94002. 415/598-2290.

SUBSCRIPTION INQUIRIES
PC TECH JOURNAL, P.O. Box 2968, Boulder, CO 80321. Subscription service: 800/525-0643, 303/447-9330. Back issues: send $7/copy ($8 outside U.S.) to Ziff-Davis Publishing, One Park Ave., 4th floor, New York, NY 10016.

PC TECH JOURNAL (ISSN 0738-0194) is published monthly, $29.97 for one year, $52.97 for two years, $69.97 for three years. Additional postage $6 for Canada & Foreign by Ziff-Davis Publishing Company, a division of Ziff Communications Company, One Park Avenue, New York, NY 10016. Second-Class Postage paid at New York, NY and at additional mailing offices. POSTMASTER: Send address changes or subscription inquiries to P.O. Box 2968, Boulder, CO 80321.

BPA SELECTED MARKET AUDIT DIVISION

1985 AWARD FOR BEST COMPUTER MAGAZINE Computer Press Association

Save On New Model of Periscope

"A marvelous debugging tool..." says Peter Norton.

SUMMER SALE.
You don't have to spend thousands, or even hundreds, of dollars to buy a high-performance debugger. Through August 31, the new software-only Periscope II-X is yours for only $95! Periscope II, normally $145, is only $125!

NEW MODEL.
Periscope I includes both a break-out switch for "spontaneous" debugging and a RAM board to protect the debugger from programs that overwrite memory. Periscope II includes a break-out switch that doesn't require an extra slot to install. And the new Periscope II-X is Periscope II without the break-out switch. So you can now get Periscope with no hardware at all!

NEW RELEASE.
The new version 2.1 enhances all models with many of the features users have requested. Call for details.

CHECK OUT THE REVIEWS.

- *"Periscope strikes a balance of power and features that will please serious PC programmers."* — Jeff Duntemann, PRODUCT OF THE MONTH, *PC Tech Journal*, 1/86
- *"This symbolic debugger, with its breakout button and powerful command options, stands unrivaled for its flexibility . . . Periscope's diverse features, affordable price, and portability place it in a class by itself."* —Ward Christensen, "Breaking out with Periscope," *PC Tech Journal*, 3/86
- *"Periscope represents the finest software debugger available in its class."* — Andrew Fried, *Computer Shopper*, 4/86

Other reviews have appeared in *Computer Language* (3/86), the Boston Computer Society's PC Report (1&2/85), and *Programmer's Journal* (Vol. 3, No. 1).

I'M CONVINCED PERISCOPE IS FOR ME!

Please send me the following:

Periscope II-X ____ @ $95* ....______

Periscope II with Switch ____ @ $125* ....______

Periscope I with Board & Switch ____ @ $295..........______

Add Shipping Cost (see below) .....______

Total Order Amount.............______

Shipping-$3 UPS; $5 COD; $8 Canada; $24 Foreign
We accept qualified PO's in the U.S.
*Prices good through August 31, 1986

Ship to:
Name ______
Company ______
Address ______
City/State/Zip ______
Telephone ______
Payment: ☐ Check enclosed ☐ COD ☐ MC ☐ Visa
Card No. ______
Expires ______
Signature ______

PERISCOPE REQUIRES an IBM PC, XT, AT, Compaq, or other close compatible; 128K RAM; DOS 2.0 or later; one disk drive; and an 80-column monitor.

Take advantage of the low sale prices while they last . . . Order your Periscope today! To order, clip the coupon, or call TOLL-FREE

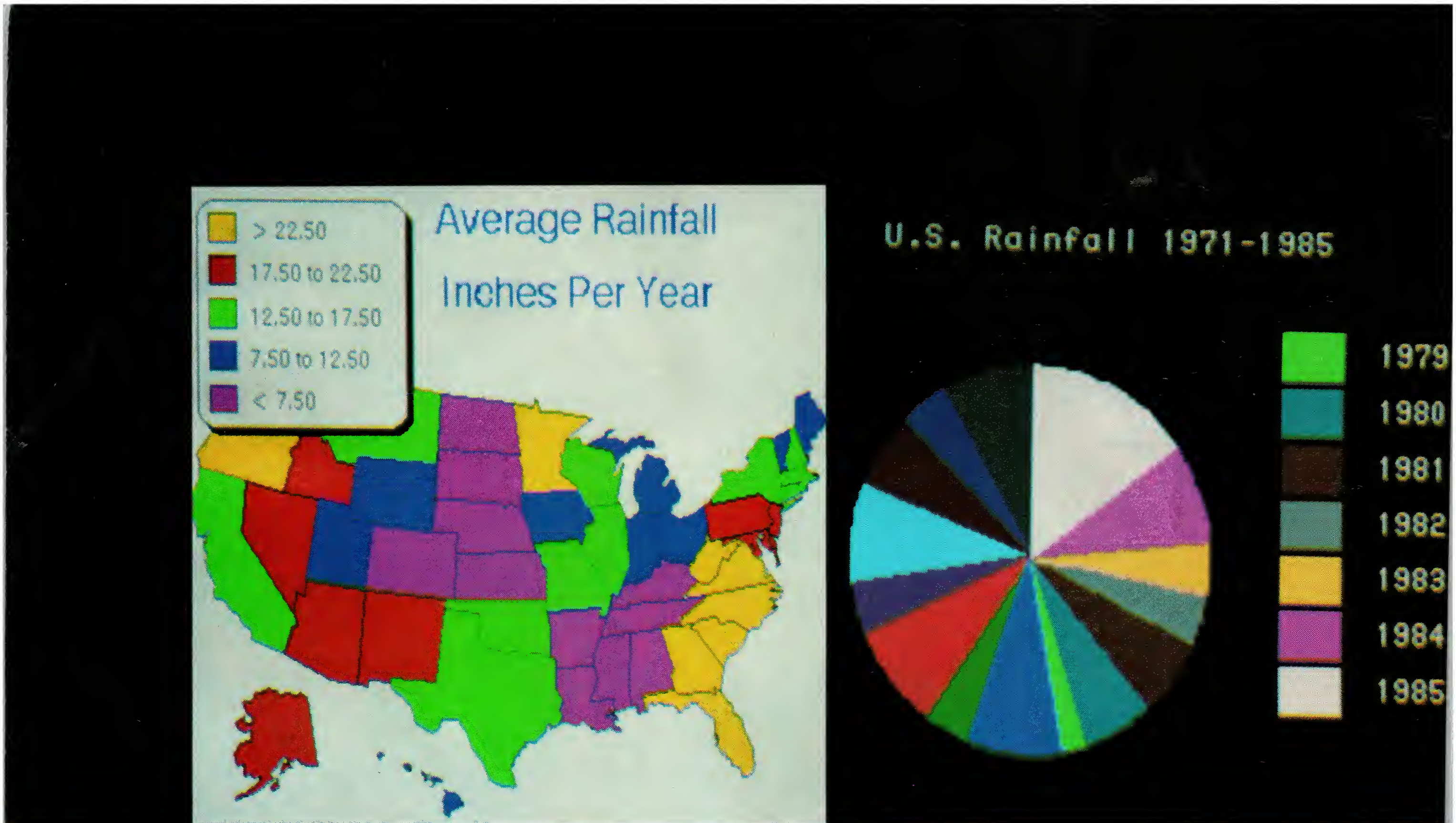
800/722-7006

30-Day Money-Back Guarantee

Get Your Programs Up and Running;
UP PERISCOPE!

Data Base Decisions
14 Bonnie Lane
Atlanta, GA 30328
404/256-3860

CIRCLE NO. 215 ON READER SERVICE CARD



EGA Mode

Plantronics Mode

Finally, An EGA Car
Responds To Yo

Introducing The Paradise AutoSwitch™ EGA Card.

You've decided you want an EGA Video Card because it offers the best quality text with high resolution color graphics. The problem is that older versions of popular software, like Lotus 1A, (and some current versions too) won't work with the EGA standard. That's a problem because in the course of your workday you'll need to use software written for different video modes.

Paradise has the answer. The Paradise AutoSwitch EGA works with every popular PC video mode. And is smart enough to switch to the right mode every time you change programs. Automatically.

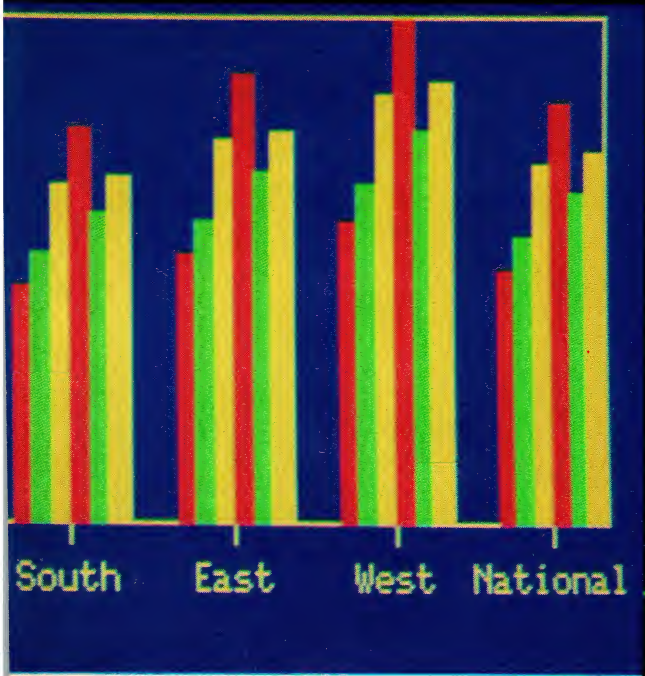
AutoSwitch. Only Paradise has it.

The Paradise AutoSwitch EGA changes video modes automatically to work with your software. No Mode commands. No special software to run. On RGB monitors it "autoswitches" between EGA, CGA (IBM Color Graphics) and Plantronics modes; on monochrome monitors between EGA, MDA (IBM Mono Text) and Hercules modes. The Paradise AutoSwitch EGA runs any PC software you want. Immediately. Automatically. That's something ordinary EGA Cards can't do.

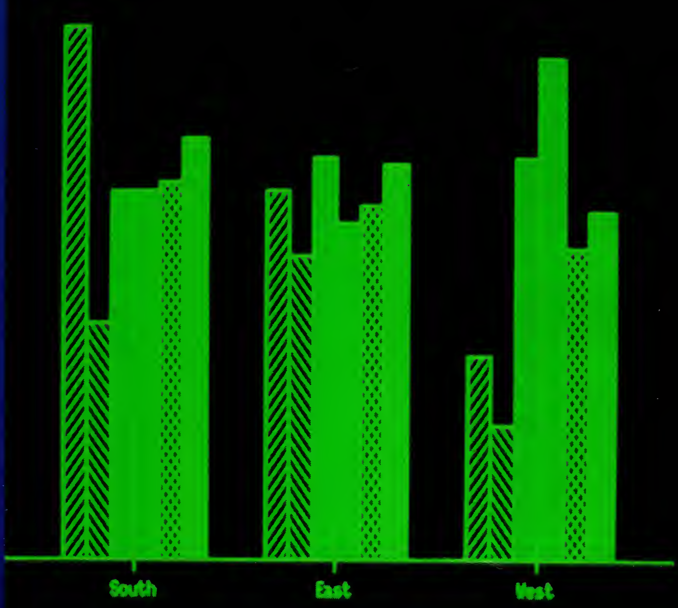
Runs all EGA, CGA, MDA and Hercules Software with true hardware level compatibility.

True hardware compatibility makes the Paradise AutoSwitch EGA Card different from other EGA Cards. No other EGA Card offers the

Trademarks: Paradise, AutoSwitch EGA Card-Paradise Systems, Inc.; Lotus 1-2-3-Lotus Development Corporation; Registered trademark: IBM-International Business Machines Corporation.



CGA Mode



Hercules Mode

d That Automatically ur Every Mode.

6845 CRT controller included on the IBM Monochrome, IBM Color and Hercules Cards. The Paradise-designed PEGA video controller chip has the equivalent of a 6845 inside. That means when the Paradise AutoSwitch EGA runs EGA software, it's an EGA card; when it runs IBM Color Graphics (CGA) software, it's a CGA card; when it runs Hercules software, it's a Hercules card. No "batch files," special software or complicated installation. No incompatibility hassles. EGA with true 6845 hardware compatibility.

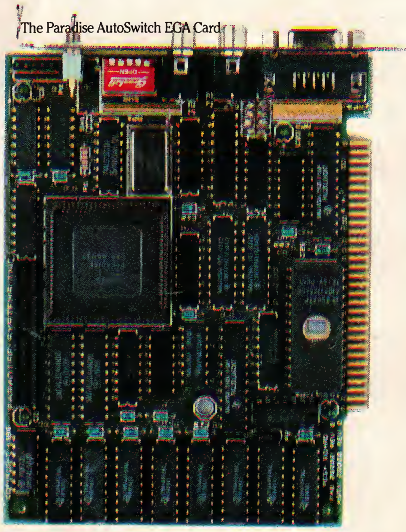
The Simpler the Better.

Paradise's exclusive AutoSwitch feature and true 6845 hardware compatibility make using the Paradise AutoSwitch EGA simple and hassle free. The Paradise-designed single chip EGA video controller makes possible all this functionality and intelligence on a short card. In fact, even the users manual is short. Only 24 pages.

The Paradise AutoSwitch EGA Card. Just what you would expect from the innovators in Video Technology.

PARADISE
SYSTEMS, INC

The Paradise AutoSwitch EGA Card



IRCLE NO. 137 ON READER SERVICE CARD

or more information call 800-527-7977 Ext. 356, toll free. From California, 800-822-2020 Ext. 356. Or stop by our favorite computer store.

ATRON BUGBUSTERS GREASE BORLAND LIGHTNING

"If I were starting a software company again, from scratch, Atron's AT PROBE™ would be among my very first investments. Without Atron's hardware-assisted, software debugging technology, the flash of Turbo Lightning™ would be a light-year away."
Philippe Kahn, President, Borland

HOW BORLAND DOES SO MUCH, SO WELL, SO FAST

We asked Borland International president Philippe Kahn to share his secrets for rapidly taking a good idea and turning it into rock-solid reality. How does the Borland team do so much, so well, so fast?

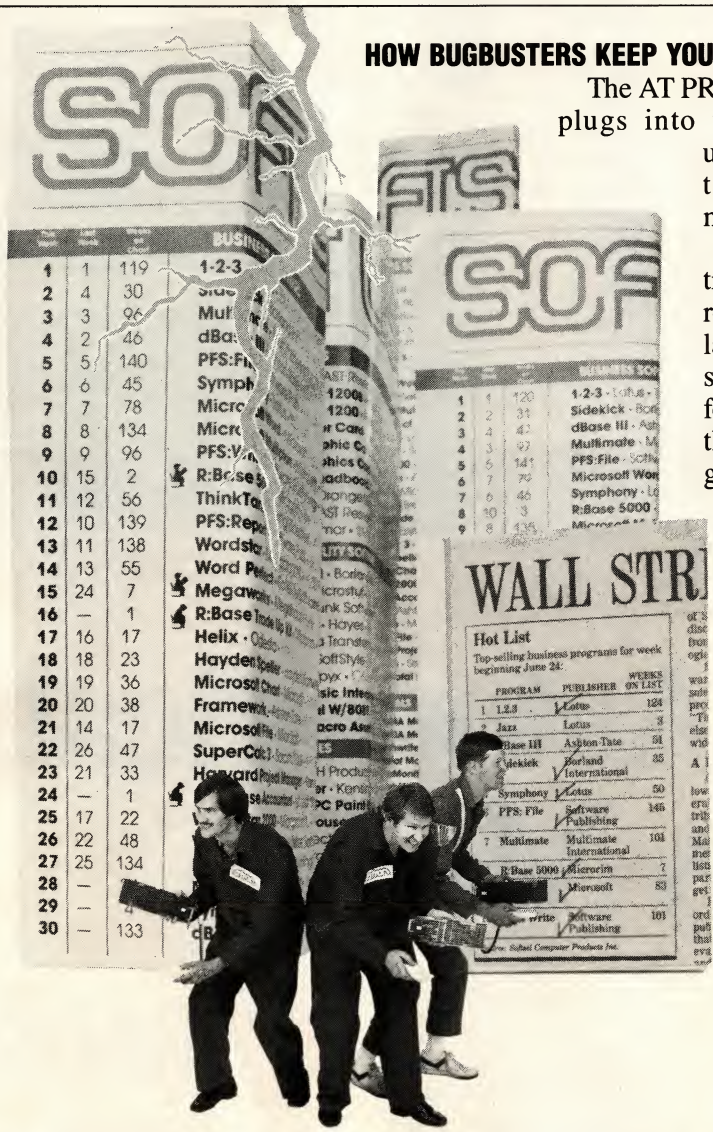
He begins, "I remember when Atron used the June 24, 1985 *Wall Street Journal* chart of top-selling software in an ad." [Note: At that time, seven of the top ten software packages were created by Atron customers; it's now now nine out of ten.] "Side-Kick was number four, and I let Atron quote me in saying that there wouldn't have been a SideKick without Atron's hardware-assisted debuggers.

"You might say lightning has literally struck again. Turbo Lightning made number four on *SoftSel's Hotlist* within weeks of its introduction! And again, I say we couldn't have done it without Atron debugging technology.

"Cleverly written code is, by definition tight, recursive, and terribly complex," he continues. "Without the ability to externally track the execution of this code, competent debugging becomes very nearly impossible."

Concludes Philippe, "And after Turbo Lightning was solid and reliable, Atron tuning software turned our Probes into performance analyzers. How do you think we *greased* our lightning?"

Philippe, along with a couple million or so of your satisfied customers, we say congratulations on yet another best-selling product. We can't wait to see what awesomely useful technology will come shooting out of Borland International next.



HOW BUGBUSTERS KEEP YOU FROM GETTING SLIMED

The AT PROBE is a circuit board that plugs into your PC/AT. It has an umbilical which plugs into the 80287 socket and monitors all 80286 activity.

Since AT PROBE can trace program execution in real time, and display the last 2048 memory cycles in symbolic or source-code form, you can easily answer the questions: "How did I get here?" and "What *are* those silly interrupts doing?"

It can solve *spooky* debugging problems. Like finding where your program over-writes memory or I/O - impossible with software debuggers.

You can even do source-level debugging in your favorite language, like C, Pascal or assembler. And after your application is debugged, the AT PROBE's performance measurement software can isolate performance bottlenecks.

Finally, the AT PROBE has its own 1-MByte of memory. Hidden and write-protected. How else could you develop that really large program, where the symbol table would otherwise take up most of memory.

LOOK AT IT THIS WAY.

History shows that non-Atron customers don't stand a very good chance of making the Top Ten list. Lightning *really does* have a way of striking twice!

The PC PROBE™ is $1595 and the AT PROBE is $2495. So call Atron today. You can be busting some really scarey bugs tomorrow. And maybe, just like Borland, you can also bust some records.

Atron
THE DEBUGGER COMPANY
20665 Fourth Street • Saratoga, CA 95070 • 408/741-5900

CIRCLE NO. 203 ON READER SERVICE CARD

Long Live the 286!

We wish the 386 would go away, at least for a while.

With this issue *PC Tech Journal* breaks from a three-year editorial policy. Until now we have been "true blue" in that we have limited our coverage of desktop computers to those manufactured by IBM. Because IBM was the majority shareholder in the market for 8088-based machines, a standard emerged that allowed *PC Tech Journal* to disperse technical information about IBM products that was perfectly usable by developers for end users of other 8086-family desktops.

IBM still dominates the market with its PC and XT models, but times have changed. In the process of bringing a new, 80286-based machine to the market and establishing yet another standard, IBM stumbled. Within three months the market responded with 286 machines of higher performance, and the race was on.

IBM still is the majority shareholder in the AT market, but it has a much smaller share than it had in the PC arena. While most estimates credit IBM with 80 to 85 percent of 8088-based PC sales, its AT share is judged to run from 65 percent to an incredible 50 percent. To pour gasoline on an already substantial fire, the AT competition usually has something significant to offer over the IBM product; this was less often the case with the PC.

The market is one consideration. *PC Tech Journal*'s readers are another, and we wanted to know what they were up to in this new market. We took advantage of the extensive research done by our advertising department and commissioned a study of our own. The results, so far, are clear.

First, our audience is moving to 286-based machines much more rapidly than the market at large. This comes as no surprise. Those who are developing software or using desktops in technical environments always want and need the best performance per dollar spent. AT-class machines deliver not only performance but also great opportunity for expansion, thus stretching the investment well into the future.

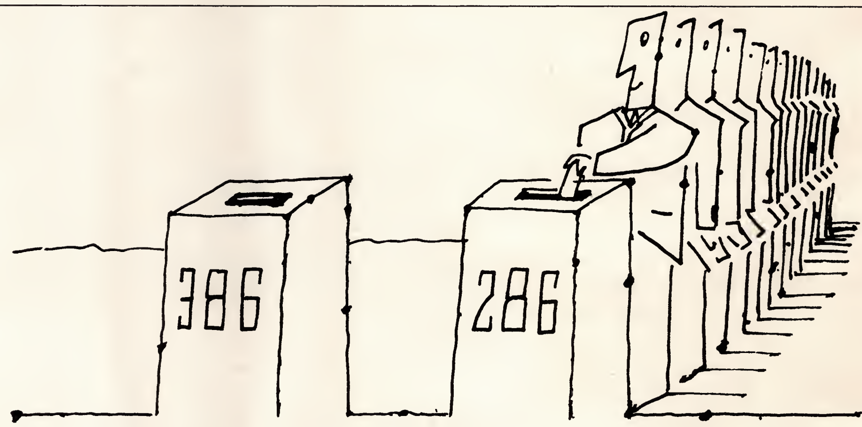


ILLUSTRATION • MACIEK ALBRECHT

Second, IBM equipment still is preferred by our readers, but just barely—50.7 percent of your companies are planning to buy IBM and of those, 58.3 percent intend to buy ATs. Another not-unexpected result: Compaq is a hearty number two with its 286 machines, and that does not even include the Portable II, which was just becoming available at the time our research was conducted.

The study indicates one more factor that is not surprising for our audience but different from the world at large. When asked which brands of AT compatibles they are considering, our readers replied with a long list, identifying almost all of the companies in the market. The less technically inclined buyer is likely to know only one or two other brands. This means that our audience, besides being more knowledgeable, is willing to exploit the advantages of alternative solutions even if it means taking a risk. Willingness to take the risk is also evidenced by the response to our question about where equipment would be purchased: an astonishing 40 percent said mail order.

All of this leads up to our decision to cover non-IBM computers in these pages, under the aegis of compatibility and performance. Our audience is certainly clear on these two points. High performance is a real draw, but not at the expense of a machine that is markedly different from the AT.

So we begin with the superb work of contributing editors Steven Armbrust and Ted Forgeron. As is usual for *PC Tech Journal*, the introductory article (see p. 52, this issue) in our series consists of a description of the methodology that will be used to evaluate each system. Armbrust and Forgeron, along with Paul Pierce, have devised a suite of programs that examines the machine's degree of compliance with the IBM specification and analyzes its performance. Bob Smith's excellent CPUID.ASM program (see "Chips in Transition," April 1986, p. 56, or call PCTECHline) was used as a starting point. The performance programs are noteworthy because they report more information than is normally considered when evaluating speed; using these programs, Armbrust and Forgeron determined that the Compaq Deskpro 286 ran its memory at 6 MHz even though the 286 was running at 8 MHz.

Appropriately, the first review in which these criteria are applied reports on the Compaq 286 machines (see p. 80, this issue). Forthcoming articles will review the HP Vectra, the TeleVideo

Tele-CAT 286, the PC's Limited 286 series, the new ITT machine, and others. With luck we will have several of these alternatives around our offices for continuing study.

386: SOONER OR LATER?

The efforts of the industry to establish standards for the construction of 386-based computers are commendable. Although IBM could ruin such attempts by announcing its own open architecture, serious work by third parties can only be beneficial. If the 386 market gets off to a good start before IBM gets into the act, IBM will have to answer for any design that strays too far from what its aftermarket (or is it beforemarket?) is prepared to support. If IBM gets in faster, perhaps the good work of outsiders will be recognized and integrated.

The 386 machines will be available soon, perhaps even by the time you read this. They will be expensive, because chips are in limited supply, but they will be interesting because the 386 has a far better memory management scheme and better support for serious operating systems than does the 286. The extent to which they will influence the desktop market in the next year is unclear; my guess is that at least 18 months will pass before the 386 desktop is available in large enough quantities to spur a serious market.

In the meantime, we are not exploiting the full potential of the 286. It is now abundantly clear that the 286 is really two chips in one: an 8086 and an 80286. In the context of DOS, it does not seem feasible to have an operating system that can run in both real and protected modes at the same time. It *is* possible, however, to have an operating system that runs in protected mode and allows 640KB applications to run if they obey a few rules and keep their hands off the hardware. That means there has to be a decision point at which a user is willing to shift from one architecture to the other. The advantage of the 80286 is that both architectures are present in the same machine, so no hardware purchases (outside of more memory) are required. What is necessary—and what is not available at the moment—is a DOS-compatible operating system that allows the transition.

The 386 offers improvements in architecture that enable an operating system to become more efficient at delivering services. The 286 offers only increased memory, with limited operating system support. These are the reasons I hear most often by those who espouse the 386 solution. Therefore, they say, ignore the 286 and migrate now.

I disagree. Most users have three simple areas in which they desire greater performance and capacity. The first is raw processor power. There is little doubt that the 386 delivers more processing power than the 6-MHz AT; even 8-MHz systems appear sluggish by comparison. However, 10- and 12-MHz 286 systems are available today that offer speeds closer to that of the 16-MHz 386. When the 20-MHz version of the 386 becomes economically attainable, the 286 will become less desirable—but for now it is acceptable.

The second area is main memory, both for executing large programs and keeping large amounts of data resident. The 386 dramatically improves the memory situation over the 286 with its better memory management unit, but most users should be able to work effectively in the 286's 16MB for the next two to three years.

The third area is disk performance. The problem with most IBM systems and their clones is that faster disks chew up more direct memory access time and thus reduce the cycles available for CPU operation. This problem is inherent in the design of the disk controller and main memory, not the processor itself. When main memory is dual-ported, with the I/O bus on one side and the CPU on the other, then real improvement will be possible.

A possible fourth concern of performance buffs is the speed of graphical operations. To date, the IBM standard architecture has promoted solutions in which the processor did all the work; as the resolution and color capability of these subsystems rises, so does the number of CPU cycles necessary to service them. The 386 offers no significant improvement over the 286 for supporting graphical operations. Clock rate improvements do help, but the number of bits that must be moved is rising faster than the performance of the base processor. The only real solution is a graphics adapter board with its own processor (such as Intel's 82786 or TI's 34010). That seems a likely direction for growth, and it is applicable to any processor, even the now lowly 8088.

As far as I am concerned, that adds up to a lot of life left in the 80286 machines—the ones that are especially attracting the attention of the *PC Tech Journal* audience and that we will all have bought so many of by the time a viable, cost-effective 386 machine is available. What we need are some new and improved hardware subsystems and, especially, a protected mode version of DOS to exploit the in-place memory capability.

This is an election year. Let's vote NO for a 386 until our investment in 286 machines has been returned, or until they have been fully depreciated, whichever comes first. And let's vote YES for expansion to the outer limits of 286 functionality.

Please drop me a postcard with your vote on it.

A NOTE ABOUT RESEARCH

Most magazines conduct regular research within their audiences. The reasons are varied, but include analysis of advertising, editorial satisfaction, future editorial directions, and general market analysis. Many of you have been called or asked to return a questionnaire. From several of your comments, I know some of you believe that our motives are selfish, designed solely to sell advertising.

We do have to sell advertising to bring you this product every month. Our advertisers need the information our research gathers not only to decide about media purchases but also to support their own market conclusions. We make no secret of that. However, you might not realize some other aspects about our research.

First, all research reports are used by the editorial staff. We contribute to almost every survey during its development to be sure it generates information helpful to us. The results are extremely valuable to us in assessing your interests and needs and the state of the market. Second, some surveys are done purely for editorial reasons and have nothing to do with selling ads. Because of the way surveys are constructed, it may be difficult to sense this as you respond. Finally, not all the surveys identify *PC Tech Journal* as the source. For reasons of statistical validity, it is sometimes necessary to conceal our identity.

I know your time is valuable. All of us appreciate your help with our research, your time permitting. What we try to offer in return is an improved *PC Tech Journal*, a magazine that better serves your needs.

—*WF*

"One of the primary reasons the Computer Press Association was formed was to promote high standards of writing in computer journalism. It's time to reward those who do it best."

Hal Glatzer
Vice President
Computer Press Association


Computer Press
Awards
1985
Best Computer Magazine
PC Tech Journal
CITIZEN
Computer Press
Association

Best Computer Magazine

We are proud and honored to have been selected as the "Best Computer Magazine" by the Computer Press Association in its first annual awards.

At the awards ceremony, it was said that "PC TECH JOURNAL has substance, style and clear writing combined with superior information, value and contemporary graphics."

Indeed, it would be difficult for us to express more succinctly the standards we set for ourselves in publishing PC TECH JOURNAL. And we accept with appreciation the fact that our professional colleagues have recognized our achievement.

By concentrating on the rapidly growing need for information about PC systems, PC TECH JOURNAL has created a unique publication for PC systems experts, the key segment of an extremely sophisticated market.

This award as the "Best Computer Magazine" commits us even more toward the development of PC TECH JOURNAL as the information source for those dedicated systems professionals whose vision and industry are fast making the personal computer the most important business tool ever invented.

INTELLIGENT COMMENTS

Thank you for the excellent review of Q-PRO 4 by Chris Christian ("A Data Manager for Intelligent Screen Forms," January 1986, p. 126). We appreciate the work reflected in the depth and accuracy of the review. We, along with all manufacturers, have convinced ourselves that our products have only good points. Nonetheless, in all key areas, we agree with your analysis of the good and bad points of the product.

The reviewer made the astute observation that Q-PRO 4's file handler, although suitable for an eight-bit environment, was not fast enough for large files under DOS. The review also mentioned that a new file handler was to be available soon. Indeed, the new B+ tree file handler was released in February 1986. It has proven to increase the product's file-handling speed in most cases by an order of magnitude.

Again, our organization's sincere appreciation to Mr. Christian and the staff for a job well done.

Robert W. Roth
President
Q-N-E International

STATISTICAL STATIC

I am generally pleased to see articles on statistics in computer magazines because the personal computer is an ideal environment for interactive statistical analysis. That is assuming, of course, the user has the appropriate software available and sufficient training in statistics to use it correctly.

I was, therefore, distressed to see the substantial errors included in the article "Statistical Correlation" by Thomas Madron in the April 1986 issue of *PC Tech Journal* (p. 126). I point out the following examples.

The caption to figure 1 states, "The correlation coefficient is the slope of the 'best fit' straight line through these points." In fact, the correlation coefficient equals the slope of the line times the ratio of the standard deviations of the two variables.

On page 128 is the statement, "A coefficient of ±1.0 implies a completely causal relation between two variables..." Of course, a unit correlation implies only that two variables are perfectly associated and says nothing about causal relationships. This is an extremely significant distinction.

The discussion of the effect of missing data on page 130 is obscure. For example, the statement "a correlation coefficient based on these two variables can have a somewhat different meaning than if all respondents had answered both questions" is meaningless since the correlation is computed on the sample of observations with data present on both variables.

The description provided for figure 5 is incorrect and incomplete. The title "Sample Correlation Matrix" is wrong—figure 5 is a contingency table display of the frequencies of occurrence of the responses to the two questions. While the rows and columns of figure 5 are never described, the text implies that 3 represents missing data and the valid responses are 1 and 2. In that case, Pearson's product-moment correlation is entirely inappropriate to describe the association between two dichotomous variables.

On page 130, the statement "CORL.FOR is a linear analysis, finding a linear least-squares fit and performing a linear transformation to normalize data around 0" is incorrect—neither correlation nor regression normalizes the data about anything. To normalize the data around 0, the user could subtract the mean and divide by the standard deviation to transform each observation to a standard normal deviate.

On page 140 (within listing 3, STAT.FOR) is the comment "Programs that calculate significance tests usually need an estimate of the number of observations. Subsequent programs use the *lowest* number of observations taken from the lower diagonal matrix as a conservative estimate since any significance tests based on a data matrix with missing data are suspect." Nonsense. In the first place, one does not estimate the number of observations since one can count them exactly. What the author probably meant to say was that in making multivariate tests of an hypotheses with missing data, some adjustments may be required to the degrees of freedom for the particular test. However, univariate tests of significance on individual correlations with different numbers of observations are entirely appropriate and valid.

In addition, while I have not examined the listings in detail, I strongly recommend that readers not use these programs until they have been validated on some multivariate data sets for which the results are known. My concern is based on the absence of a discussion of the substantial numerical problems involved in multivariate statistical computing. The author even failed to take the precaution of performing the operations on sums and sums of squares in double precision. The potentially disastrous results due to accumulating round-off errors in single precision (REAL*4 type on page 140) should not be underestimated.

If you expect your magazine to have any credibility at all in the statistical community, you must take stronger action to ensure that your editorial process keeps such erroneous information from your readers.

David N. Ikle, Ph.D.
Denver, CO

It is gratifying to see interest in statistical applications such as Thomas Madron's article "Statistical Correlation." Unfortunately the choice of algorithm used to computer the correlations (formula *c* on page 129) is not very accurate when applied to large numbers

and/or large amounts of data. Several alternative algorithms have the desirable property of requiring only one pass through the data without incorporating a final subtraction of very large numbers where much precision is lost. One such algorithm is described below.

Suppose I were intercorrelating K variables based on data from N observations, and X_{ij} is the value on variable j from observation i. First, initialize the variable sums

$$m_1^{(0)} \ldots m_K^{(0)}$$

to 0 and the lower triangular matrix

$$v_{ii'}^{(0)} \text{ for } i = 1 \ldots K,\ i' = 1 \ldots i$$

to 0. Now, at each step, as $j = 1$ to N ranging over the N observations, I would accumulate the following:

$$m_i^{(j)} = m_i^{(j-1)} + X_{ij} \text{ for } i = 1 \ldots K$$

in addition to

$$v_{ii'}^{(j)} = v_{ii'}^{(j-1)} + \frac{(jX_{ij} - m_i^{(j)})\ (jX_{i'j} - m_{i'}^{(j)})}{j(j-1)}$$

for $i = 1 \ldots K$ and $i' = 1 \ldots i$

The second step shown above is skipped over when $j = 1$. After j reaches N, the mean for variable i is

$$m_i^{(N)}/N$$

the standard deviation is

$$\sqrt{\frac{v_{ii}^{(N)}}{N - 1}}$$

and the correlation of the two variables i and i' is as follows:

$$r_{ii'} = \frac{v_{ii'}^{(N)}}{\sqrt{v_{ii}^{(N)} v_{i'i'}^{(N)}}}$$

One other small error (which does not affect the correlations) is the use of N rather than N − 1 in the denominator when calculating the standard deviation (p. 132). N is correct when calculating the standard deviation over an entire population. When analyzing a sample from the population (as in almost all applications), the use of N underestimates the population value.

Alan Bostrom, Ph.D.
President
Crunch Software Corporation

Some of the comments by both Mr. Ikle and Mr. Bostrom are correct and well-taken. Mr. Bostrom particularly made several thoughtful points that should be considered when doing statistical programming. Unfortunately, several of the problems that were identified by Mr. Ikle crept into the article during the editorial process. The points on which he is clearly correct—his comments about unit correlation, figure 5, and CORL.FOR—were editorial glitches.

Of more substance are the comments by Mr. Bostrom when he observes that the choice of algorithm can affect the numerical accuracy of the calculations. An example of this can be found by compiling the correlation program in single precision without an 8087 math coprocessor, then running it with data that are ill conditioned, such as the so-called Wampler data set, part of which is reproduced as follows:

200000	1.00001	0.99999
600000	2.00001	1.99999
400000	1.99999	2.00001
400000	0.99999	1.00001

The resulting calculation will generate a −4.0 correlation between variables 1 and 3—clearly a ridiculous result (the correct answer is 1.0).

Mark Williams

Now the biggest name in C compilers comes in a size everybody can afford.

Let's C.™

$75

Introducing Mark Williams' $75 C compiler. Want to explore C programming for the first time? Or just on your own time? Now you can do it in a big way without spending that way. With Let's C.

This is no little beginner's model. Let's C is a powerful programming tool, packed with all the essentials of the famous Mark Williams C Programming System. The one chosen by Intel, DEC, Wang and thousands of professional programmers. The one that wins the benchmarks and the reviewers' praise:

"(This compiler) has the most professional feel of any package we tested..."—BYTE

"Of all the compilers reviewed, (it) would be my first choice for product development."—David W. Smith, PC WORLD

And now for more big news. Get our revolutionary csd C Source Debugger for just $75, too. You can breeze through debugging at the C source level ignoring clunky assembler code.

Affordable, powerful, debuggable. Mark Williams Let's C is the big name C compiler at a price you can handle. Get your hands on it now.

Mark Williams Let's C
- **For the IBM-PC and MS-DOS**
- **Fast compact code plus register variables**
- **Full Kernighan & Ritchie C and extensions**
- **Full UNIX™ compatibility and complete libraries**
- **Small memory model**
- **Many powerful utilities including linker, assembler, archiver, cc one-step compiling, egrep, pr, tail, wc**
- **MicroEMACS full screen editor with source**
- **Supported by dozens of third party libraries**
- **Upgradeable to C Programming System for large scale applications development**

Let's C Benchmark Done on an IBM-PC/XT, no 8087. Program: Floating Point from BYTE, August, 1983.

	Exec Time in Seconds
Let's C	134.20
MS 3.0	347.45

Use this coupon or charge by calling toll-free: 1-800-MWC-1700. In Ill. call 312-472-6659.

ORDER NOW! 60-DAY MONEY BACK GUARANTEE!

Mark Williams Let's C

Please send me:

_____ copies of Let's C and _____ copies of csd (C Source Debugger) at $75 each. (Ill. residents add 7% sales tax.)

☐ Check ☐ Money Order ☐ Visa, MasterCard or American Express

Name_______________

Address_______________

City_______________ State_____ Zip_____

Card #_______________ Exp. Date_______

Signature_______________

TJ0886

Mark Williams Company
1430 West Wrightwood
Chicago, Illinois 60614

CIRCLE NO. 205 ON READER SERVICE CARD

LETTERS

A change in algorithm cannot, by itself, correct problems generated by very problematic data. That may be accomplished in two ways: through the use of double precision or by the use of an 8087 coprocessor (or both). An executable file containing the compiled correlation program—compiled to take advantage of the 8087—is available on PCTECHline. As a side note, I would suggest that serious statistical or other mathematical programming on a microcomputer must assume the availability of an 8087. With the 8087, program CORL produces results for the Wampler data that are essentially the same as major mainframe statistical packages, such as SAS. A combination of an 8087 and double precision may be required just to read and write the data properly. You should be aware, however, that most real data are not so ill conditioned as the Wampler data.

Mr. Bostrom also notes that I calculated the standard deviation for a population rather than a sample. Very true. Users may wish to change the formula for calculating the standard deviation in subroutine CORR (p. 140). Within that subroutine, the following set of lines appears:

```
      DO 80 I = 1,NV
         FMEAN(I) = FMEAN(I)/ T
         STD(I) = SQRT(R(I,I)/ T-FMEAN(I)**2)
80    CONTINUE
```

The line that begins with STD(I) = is changed so that the loop becomes:

```
      DO 80 I = 1,NV
         STD(I) = SQRT((R(I,I)-FMEAN(I)**2/T)
      Z/(T-1.0))
         FMEAN(I) = FMEAN(I)/ T
80    CONTINUE
```

This is one among several algorithms that will accomplish the desired results.

With the foregoing comments said, I will address the remaining criticisms from Mr. Ikle. On the first point, Mr. Ikle is only partially correct. It is true that one way to define the correlation coefficient is as he states it. If however, correlation coefficient is defined in terms of "standard" scores (or standard normal deviates)—where both variables have been transformed to have a mean of 0 and a standard deviation of 1—and where the slope of the regression is based on standard scores, then the definition as given is correct.

Another common definition is that the correlation coefficient is the ratio of the covariation to the square root of the

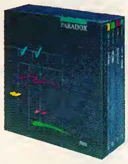
FOUR STARS

★★★★

Paradox® is the only relational database ever to earn this honor from Software Digest.

In all, 15 relational databases were tested. And when the results were tabulated, Paradox was the only one to win high ratings in every category. From ease of learning through ease of use, error handling, performance and versatility.

So not only was it Software Digest's number one relational database, it's now the only one ever to earn their coveted four-star rating. And if you think that's good, see our new version, Paradox 1.1. For the name of the Ansa dealer nearest you, call 800-547-3000, Dept. 154; in Oregon, 503-684-3001.



PARADOX

Relational Database by Ansa Software

Software Digest Rating	Overall Evaluation	Program Name	Version Tested	Ease of Learning	Ease of Use	Error Handling	Performance	Versatility	Memory Requi	Price	
★★★★	8.1	Paradox	1.0	green	green	green	green	green	512K	$695	✓
★★★	7.8	R:base 5000	1.01	yellow	green	green	green	green	320K	$700	✓
★★★	7.3	The Smart Data Manager	2.00	yellow	yellow	green	green	green	256K	$495	✓
★★★	7.1	dBase III Plus	1.0	yellow	yellow	yellow	green	green	256K	$695	✓
★★★	7.0	PractiBase	1.14	green	green	yellow	yellow	green	256K	$70	✓
★★★	7.0	Dataease	Pre. 2.5	yellow	yellow	yellow	yellow	green	384K	$600	✓
★★	6.9	Power-base	2.10	green	green	green	red	green	320K	$349	✓

green 7.0-9.9 yellow 5.0-6.9 red under 5.0

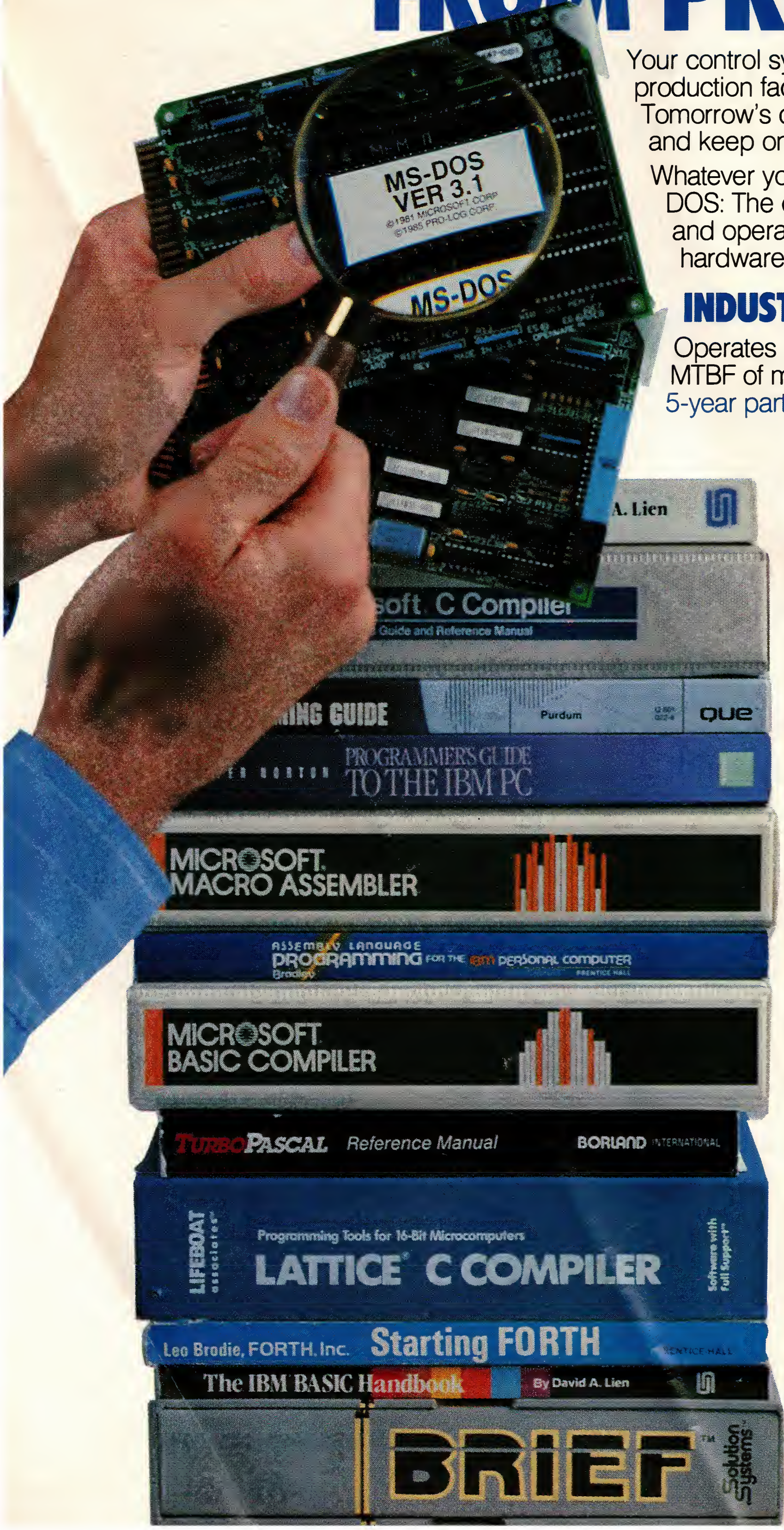
Software Digest **RATINGS NEWSLETTER**

Test results reprinted from cover of Software Digest March 1986.
Software Digest is a registered trademark of Software Digest, Inc., Philadelphia, Pa.

Paradox is a registered trademark of Ansa Software.

CIRCLE NO. 198 ON READER SERVICE CARD

MS-DOS 3.1 ON THE STD BUS: INDUSTRIAL QUALITY AND COMMERCIAL CONVENIENCE FROM PRO-LOG.



Your control system has a lot riding on it. A million-dollar production facility. Life-giving medical equipment. Tomorrow's daily newspaper. It needs to work. . . and keep on working.

Whatever your application, you can benefit from Pro-Log's STD-DOS: The convenience and familiarity of IBM PC/AT® software and operating standards, but in rugged, industrial-quality hardware. It works . . . and keeps on working.

INDUSTRIAL QUALITY:

Operates 0° to 65°C for more than 10 years with an average MTBF of more than 20 years per card. Backed by Pro-Log's 5-year parts and labor warranty.

COMMERCIAL CONVENIENCE:

Microsoft's® MS-DOS 3.1 is a software bus for connecting libraries, languages, editors, and application programs.

The STD BUS is an industry standard hardware bus for connecting memory, I/O, and peripheral functions.

The IBM PC/AT or clone is your development tool.

STD-DOS FEATURES: WITH 2 CARDS

- 8088 CPU, 5 MHz
- Time-of-day clock
- MS-DOS 3.1 in ROM disk
- 128K-byte user ROM disk
- 16K-byte user RAM disk
- 112K bytes system RAM
- RS-232-C, RS-422/485 ports
- I/O expandable to 24 STD BUS card slots
- Memory expandable to 640K bytes
- Under $1200

NEW SOFTWARE! . . . STD-LIB 1.1: a fully documented disc-resident library of driver software that supports Analog and Digital STD BUS cards for "C", BASIC, and Assembly language programs.

For information on STD-DOS products, local seminars, or a personal demonstration, CALL or WRITE TODAY.

PRO-LOG® CORPORATION

Circle Reader Service No. 252 for Literature, No.253 for Direct Factory Contact
2560 Garden Road, Monterey, CA 93940 (408) 372-4593, Outside CA: (800) 538-9570, TLX: 171879; Australia (3) 836-3533; Canada (416) 625-7752; England (0276) 26517; France (1) 3956-8142; Germany (07131) 50030; Italy (2) 498-8031; Switzerland (01) 62 44 44.

product of the variation in X *and the variation in* Y. *This is the definition Mr. Bostrom used in his suggestion for an alternative computational algorithm. The point is that several definitions exist for the correlation coefficient that ultimately can be shown to be algebraically equivalent, although not necessarily equivalent in computational precision.*

Mr. Ikle's comments on the consequences of missing data result primarily from a difference of opinion on how extensively an issue can be discussed in a short article. My statement is not meaningless. Missing data results in restricting the range of values (variation) on one or both variables. The consequence of this is that the maximum value of the correlation coefficient may not be 1.0. In a correlation matrix, therefore, where the total number of observations may vary from one correlation coefficient to another, they may have varying maximum values. When the sample size is large and the number of missing values is small, these differences are inconsequential. In other words, sampling over a small and restricted range for one variable has the effect of decreasing the value of r. *I do not think a complete explanation of the implications of missing data was within the scope of the article.*

Concerning comments that appear in listings, they are placed in the code simply to provide a warning, not as a full explanation of the problems of calculating significance tests. Mr. Ikle is obviously correct in his comments on univariate tests of significance, but the fact is that virtually all tests of significance are sensitive to sample size and in multivariate statistical tests, varying sample sizes for difference correlations in the matrix will present problems of interpretation. A rigorous approach would be to restrict tests of significance only to situations in which all data were present or to calculate statistics only on complete data matrices.

Dealing with statistics in a short article is difficult at best. It is not possible to discuss every pitfall—that is the task of statistics text books or more scholarly articles. Perhaps some of these issues can be dealt with in future articles.

—Thomas Madron

WILL THE REAL BUG...

Your article "Taking Command in Turbo Pascal" (Programming Practices, David D. Steiner, March 1986, p. 161) has an interesting bug. The author attempts to work around a bug in DOS 2.0, but fails. The bug is that after a

DOS EXEC function call (4BH), the stack pointer registers SS and SP are destroyed. The only registers that can be counted on after returning from an EXEC call are CS and IP. Therefore it is necessary to save SS and SP in CS-relative storage before calling EXEC.

Because the EXEC-calling routine, DOS4BH.ASM, is written as a Turbo Pascal external routine, it cannot be assumed to be at any fixed origin within the code segment. A trick is needed to access CS-relative storage. A NEAR CALL is made to the immediately following instruction. POPping BX retrieves the address from which the call was made, giving the program an offset that can be used to locate CS-relative data. However, if the DOS bug has occurred, that CALL instruction pushes a word of data onto a stack that is not valid. This destroys two bytes of data *somewhere* in memory. Where, is anyone's guess. Thus, the obvious DOS bug has been exchanged for a far more subtle and hard-to-find bug in the calling routine.

I suspect that DOS4BH.ASM and the rest of the routines in the article were tested only under DOS versions 2.1 and later. Because the DOS bug is not present in those versions, the mistaken work-around will cause no problems: the stack is valid when the CALL is made, and no damage is done.

It would seem that EXEC cannot be called reliably from Turbo Pascal. But that is not so. The solution is to write the EXEC-calling routine as an in-line statement instead of an external. Turbo Pascal will generate the address of a typed constant (CS-relative initialized variable) into an in-line statement. That address is correct as is; it requires no tricks to properly locate the data. Therefore, no stack manipulations are required during the critical period while the stack is invalid, and the routine will work on DOS versions 2.0 to 3.1.

Bela Lubkin
Santa Cruz, CA

I read with interest David D. Steiner's article "Taking Command in Turbo Pascal" and also Bela Lubkin's letter (over CompuServe) pointing out the bug in DOS4BH.ASM, included with the article. While I agree with Mr. Lubkin that in-line code is preferable to an external routine when invoking EXEC from a Turbo Pascal program, DOS4BH.ASM can be modified to correct the bug. Replace the four MOV instructions that save SS and SP with the following two:

```
MOV CS:LOADSS+1[BX],SS
MOV CS:LOADSP+1[BX],SP
```

"Turbo Pascal... is a very good system. But don't make the mistake of trying to use it for large programs."

Niklaus Wirth
as reported in Micro Cornucopia
August-September 1985

LOGITECH

MODULA-2/86

and replace the seven instructions from CALL RELO2 through MOV SP, DX, which restore SS and SP, with:

```
LOADSS: MOV SP,0
        MOV SS,SP
LOADSP: MOV SP,0
```

These changes store the values of SS and SP directly into the move immediate instructions that will restore SS and SP after the EXEC call. The values thus stored replace the dummy 0 values in the move immediate instructions above.

Self-modifying code is normally to be avoided in most situations.

Randy Forgaard
Watertown, MA

Mr. Lubkin's comments about the subtleties of executing subprocesses with DOS 2.0 compared to versions 3.x are substantially correct, although the bug he identifies may not need correcting.

He correctly notes that when DOS4BH regains control from the subprocess and uses a local CALL and POP to calculate a relocation factor, a word is PUSHed onto the stack before the Turbo Pascal stack is restored. If DOS 2.x is used, then this stack will not be the Turbo Pascal stack. Mr. Lubkin is incorrect that this can overwrite a word anywhere in memory, because the value of the SS:SP register pair following a DOS EXEC call is not random. Rather, it points to the region that DOS last used as a stack. Hence, it is improbable that harm will befall either a resident user application or the operating system.

I have modified DOS4BH to eliminate the "bug." Before the relocation factor is recalculated, the SS:SP pair is set to 0:03C2H, a word of memory that the IBM Technical Reference *says is used by the "BASIC interpreter while BASIC is running." This assures that the word modified by the Call is unneeded. (DOS4BH.ASM is on PCTECHline.)*

Mr. Forgaard's solution is elegant, but self-modifying code should *be avoided. Also, his changes will not assemble with Microsoft MASM 3.0. His first two MOVs produce "Illegal Item Size" messages. Replace them with:*

```
MOV WORD PTR CS:LOADSS+1[BX],SS
MOV WORD PTR CS:LOADSP+1[BX],SP
```

—David D. Steiner

ERRATUM

In "Matching Regular Expressions" (Programming Practices, Jon Forrest, May 1986, p. 191), note that program listings 2 and 3 were transposed. *PC Tech Journal* regrets the error.

A few good reasons

WORDPERFECT
WORDPERFECT CORPORATION
DBASE III PLUS
ASHTON-TATE
EASY BUSINESS SYSTEM PLUS SERIES
COMPUTER ASSOCIATES MICRO PRODUCTS DIVISION (SORCIM/IUS)
MICROSOFT WORD
MICROSOFT CORP.
OPEN SYSTEMS ACCOUNTING SOFTWARE
OPEN SYSTEMS, INC.
HARDISK ACCOUNTING SERIES
GREAT PLAINS SOFTWARE
R:BASE 5000
MICRORIM
NETWORK REVELATION
COSMOS
WORDSTAR 2000
MICROPRO INTERNATIONAL
MULTIMATE ADVANTAGE
MULTIMATE INTERNATIONAL CORP.
DATAFLEX
DATAACCESS CORP.
MULTIMATE PROFESSIONAL WORD PROCESSOR
MULTIMATE INTERNATIONAL CORP.
MDBS III
MICRO DATA BASE SYSTEMS, INC.
KNOWLEDGEMAN
MICRO DATA BASE SYSTEMS, INC.
SOLOMON III
TLB
MBA ACCOUNTING SOFTWARE
MICRO BUSINESS APPLICATIONS
EXCALIBUR PLUS SERIES
ARMOR SYSTEMS INCORPORATED
INFORMIX-SQL
RELATIONAL DATABASE SYSTEMS
SAMNA WORD III
SAMNA
CYMA GENERAL BUSINESS
CYMA/MCGRAW HILL
ACCOUNTING SOFTWARE
LIBRA PROGRAMMING
DBMAN-NET
VERSASOFT CORPORATION
SAMNA+
SAMNA
FASTRAK
CORE SOFTWARE, INC.
ZIM
ZANTHE
LAN: DATASTORE
SOFTWARE CONNECTIONS
INFORMA
UNLIMITED PROCESSING, INC.
INTEGRATED ACCOUNTING SOFTWARE
PROMARK LTD.
TEAM-UP
UNLIMITED PROCESSING, INC.
B.O.S.S.
AMERICAN PLANNING CORP.
METAFILE
METAFILE INFORMATION SYSTEMS, INC.
OFFICE WRITER
OFFICE SOLUTIONS
Q-PRO 4
QNE
THE BIG 8 ACCOUNTING
MODTECH
GOLDATABASE
GOLDATA COMPUTER SERVICES, INC.
THE SENSIBLE SOLUTION
O'HANLON COMPUTER SYSTEMS
MICRO-ADAPT
DATA FINANCIAL SYSTEMS
EXCALIBUR SERIES
ARMOR SYSTEMS
BBX-BUSINESS BASIC EXTENDED
BASIS INCORPORATED
CLEO 3270 BSC CLUSTER
CLEO SOFTWARE
HIGGINS ADMINISTRATIVE SOFTWARE
CONETIC SYSTEMS INCORPORATED
NETWORK COURIER
CONSUMERS SOFTWARE, INC.
HARDISK ACCOUNTING SERIES
GREAT PLAINS SOFTWARE

TOPVIEW
IBM
SMART SOFTWARE SYSTEM
INNOVATIVE SOFTWARE CORP.
MBP COBOL
MBP SOFTWARE AND SYSTEMS
WORDSTAR 2000 PLUS
MICROPRO INTERNATIONAL
CC:MAIL
PPC/SYSTEMS
ACCOUNTS PAYABLE
REALWORLD CORP.
GENERAL LEDGER
REALWORLD CORP.
JOB COST
REALWORLD CORP.
INVENTORY CONTROL
REALWORLD CORP.
RM/COBOL
RYAN-MCFARLAND CORPORATION
BTRIEVE/N
SOFTCRAFT, INC.
OPEN ACCESS 2
SOFTWARE PRODUCTS INTERNATIONAL, INC.
DBIII COMPILER
WORDTECH SYSTEMS, INC.
MARKET MANAGEMENT SYSTEM
ABACUS
PRISMA PRODUCTION
ABC SYSTEMS AG
C-MAIL/C-MERGE
ACCESS DYNAMICS, INC.
CENTRAL SERVICES MGMT SOLUTION
ADVANCED HEALTHCARE SYSTEMS, INC.
HOT PRINT
ADVANCED PC SOLUTIONS
MARCON II
AIRS, INC.
AMBER
AMBER SOFTWARE CORPORATION
P.O. WRITER
AMERICAN TECH, INC.
COLLECT
ANOTHER DIRECTION DATA
HPMS
ANOTHER DIRECTION DATA
H.I.S.
ANOTHER DIRECTION DATA
RAISE
ANOTHER DIRECTION DATA
PHYSICIANS BILLING
ANOTHER DIRECTION DATA
CUMS
ANOTHER DIRECTION DATA
PROJECT WORKBENCH
APPLIED BUSINESS TECHNOLOGY
TELECONFERENCING
APPLIED COMPUTER RESOURCE SERVICES
JOB TRACKING
APPLIED COMPUTER RESOURCE SERVICES
TEN-IN-ONE
ARMOR SYSTEMS
MAINSTREET MERCHANDISER
ARMOR SYSTEMS
DONOR TRACKING SYSTEM
ARMSTRONG BUSINESS COMPUTERS
DENTALIS SYSTEM VI
ARTICULATE PUBLICATIONS, INC.
MEDCALIS SYSTEM VI
ARTICULATE PUBLICATIONS, INC.
PM/TURBO
ASPEN COMPUTER SYSTEMS, INC.
DEX
ATLANTA TECHNOLOGIES
CALL-CONTROL
ATLAS SYSTEMS
SERVICE MASTER
AVNET COMPUTER TECHNOLOGY
AUTOMATED PROOF CORRECTION SYSTEM
BACKROOM SYSTEMS GROUP
RESEARCH & ADJUSTMENT CONTROL
BACKROOM SYSTEMS GROUP
LARGE ITEMS MONITORING SYSTEM
BACKROOM SYSTEMS GROUP
RETURN ITEMS CONTROL SYSTEM
BACKROOM SYSTEMS GROUP

AUCTIONEER 3 & LISTRIEVE
BAROQUE SOFTWARE SYSTEMS
TRACTOR & IMPLEMENT DEALERS ACCOUNTING
BASIC SOFTWARE SYSTEMS
PC NETWORKING PRODUCTS
BRAEGEN CORP.
NETUTILS
BRITISH RAILWAYS BOARD
TIME ANALYSIS AND BILLING
BUSINESS APPLICATIONS SOFTWARE, INC.
CLIENT ACCOUNTING/GL
BUSINESS APPLICATIONS SOFTWARE, INC.
THE OFFICER
BUSINESS SYSTEMS RESOURCES
PM/TURBO
CALDWELL COMPUTER ENTERPRISES
CHARTERHOUSE ACCOUNTING SYSTEMS
CHARTERHOUSE SOFTWARE CORP.
CHARTER ACCOUNTING SYSTEMS
CHARTERHOUSE SOFTWARE CORP.
CB86 TO NOVELL INTERFACE
CHECKPOINT SYSTEMS
VIDWARE
CHECKPOINT SYSTEMS
AUTOBOOK
CISCO, INC.
WHEELS
CISCO, INC.
CITATIONS
CISCO, INC.
TIES
CISCO, INC.
CAD
CISCO, INC.
IMIS
CISCO, INC.
CASS
CISCO, INC.
POSSE
CISCO, INC.
ELECTRONIC PEGBOARD
COLONIAL VALLEY SOFTWARE, INC.
NOVELLIB & NOVELLIB-L
COMMUNICATION HORIZONS
PIER
COMP-TRON, INC.
EARTHWK
COMP-TRON, INC.
BEAMS
COMP-TRON, INC.
CTCOGO
COMP-TRON, INC.
STABY
COMP-TRON, INC.
PGELEV
COMP-TRON, INC.
STRAN
COMP-TRON, INC.
CADLINK
COMP-TRON, INC.
POLICY EXPRESS
COMPARE DATA CORPORATION
COMPUNEWS CLASSIFIED AD MANAGEMENT
COMPUNEWS, INC.
SCRIPMASTER/PC
COMPUTER, INC.
THE PROGRAM MANAGER
COMPUTERIZED BUSINESS SOLUTIONS
HERITAGE LIVESTOCK AUCTION
COMPUTERIZED SERVICE AND DESIGN
MENU SYSTEM
COTTAGE INDUSTRIES, INC.
4GL
CRAIGMORE HOUSE, LTD.
CYBERGRAM WIDE AREA MAIL
CYBERTEK COMPUTER PRODUCTS, INC.
MICRO INFORMATION MANAGER
CYBERTEK COMPUTER PRODUCTS, INC.
CYBERSCRIBE V
CYBERTEK COMPUTER PRODUCTS, INC.
AUTOSERV
CYLLON SYSTEMS INTERNATIONAL, LTD.

MAILING LIST
DATABASE GROUP
PROF TIME MANAGER AND BILLING
DATAFLOW
THE TERRA SYSTEM
DATASPHERE COMPUTER SYSTEMS, INC.
CLIPLOCK
DELTA CONTRACTING SERVICES, INC.
DS BACK UP
DESIGN SYSTEMS, INC.
MAIL ORDER SYSTEM
DIRECT BUSINESS SOLUTIONS
PSI/RIS,MOS/IS,MED/SIS
DIVERSIFIED BUSINESS SYSTEMS, INC.
RODE/PC DATA ENTRY SYSTEM
DPX
CUSTOM SOFTWARE
EAGLE ENTERPRISES
ATTACH/X.25 PROGRAM
EICON TECHNOLOGY CORPORATION
ACCESS/X.25 PROGRAM
EICON TECHNOLOGY CORPORATION
TRUCKING ACCOUNTING
ELROM INTERNATIONAL, INC.
COUNTRY CLUB MEMBER BILLING
ERNEST A. JONSON AND CO.
Y-MANAGER
EXECUTIVE COMPUTER TECHNOLOGIES
OMECOM
FAR WEST SYSTEMS
LOAN PROCESSING SYSTEM
FINANCIAL INDUSTRY COMPUTER SYSTEMS
MORTGAGE LOAN SERVICING SYSTEM
FINANCIAL INDUSTRY COMPUTER SYSTEMS
COMMERCIAL/CONST. LOAN SERVICE
FINANCIAL INDUSTRY COMPUTER SYSTEMS
2001 ACCOUNTING SYSTEM
FINANCIAL INFORMATION SYSTEMS
PHARMACY MANAGEMENT SYSTEM
FOUNDATION SYSTEMS
FOURTH SHIFT CORP.
FOURTH SHIFT CORP.
CPA-11
FRANKLEN COMPUTER SYSTEMS, INC.
TCB-11
FRANKLEN COMPUTER SYSTEMS, INC.
PMS-11
FRANKLEN COMPUTER SYSTEMS, INC.
GARPAC APPAREL MGMT SYSTEM
GARPAC CORPORATION
PLANT EQUIPMENT MANAGEMENT
GENERAL PHYSICS CORPORATION
SALES PROSPECT TRACKING
GOLDATA COMPUTER SERVICES, INC.
YOUR MOVE
GOLDATA COMPUTER SERVICES, INC.
SOS
GOLDATA COMPUTER SERVICES, INC.
BASIS
GOLDCOAST SOFTWARE SYSTEMS, INC.
THE DISTRIBUTOR
HIGH TECH INDUSTRIES
RESOURCE MANAGER
HIGH TECH INDUSTRIES
CITY MANAGER
HIGH TECH INDUSTRIES
HIMED
HISOFT COMPUTERS PTY., LTD.
HIPORTFOLIO
HISOFT COMPUTERS PTY., LTD.
FAMOUS
HOLM-DIETZ COMPUTER SYSTEMS, INC.
BASIC 3.2
IBM
INFOTEAM ACCOUNTANT
INFO DESIGNS, INC.
HOME CARE INFORMATION SYSTEMS
INFOMED
BCS SOFTWARE
INFORMATIC SYSTEMS

CIRCLE NO. 112 ON READER SERVICE CARD

MONITOR SYSTEMS GROUP
INFOSERV FOOD SERVICE MANAGEMENT SYSTEM
COMEAGAIN
INTERNATIONAL EXPERT SYSTEMS, INC.
NETWORK MANAGEMENT REPORT UTILITY
J.A. LOMAX AND ASSOCIATES
RETAIL FINANCIAL MGMT SYSTEM
JAHL DATA SYSTEMS
POWER PAYROLL
JCA SYSTEMS
CLASS
KLEIN & KLEIN
OMICROM MEDICAL MGMT SYSTEM
LEDON
THE ASSISTANT
LIBRARY AUTOMATION PRODUCTS, INC.
LODGMATE
LODGISTIX
MACOLA FINANCIAL SOFTWARE
MACOLA, INC.
HR-1
MAINFRAME MICROS
HR-APPLICANT TRACK
MAINFRAME MICROS
RETAIL MANAGEMENT SYSTEM
MANAGEMENT INFORMATION SOLUTION, INC.
MEGASOURCE HOSPITAL PHARMACY
MEGASOURCE, INC.
ACCOUNTMASTER/ DISTRIMASTER
METAPLUS INTERNATIONAL
INTERNATIONAL BANKINGS PRODUCT
MICRO BANKING AUTOMATION, INC.
MORTGAGE TRACKING SYSTEM
MICRO VENTURES, INC.
MULTISOFT
MULTISOFT
NCG ACCOUNTING
NCG COMPUTER SYSTEMS
N.E.T. UTILITIES
NETWORK ENHANCEMENTS TOOLS
SIMCOM HOTEL MGMT SYSTEM
NEW SYSTEMS, INC.
GLAS PAC I
PACIFIC COMPUTER & INFORMATION
NETWORKED MCBA ACCOUNTING
PACIFIC DATAWARE
PRACTICE MANAGER
PACIFIC MEDSOFT, INC.
NETWORK ASSISTANT
PACIFIC SOFTWARE
TRUST ACCOUNTING
PENTABS
DEFINED BENEFIT
PENTABS
PENSION SELECTOR SYSTEM
PENTABS
401(K)/DEFINED CONTRIBUTION
PENTABS
TIME & CHARGES SYSTEM
PENTABS
DATA MANAGEMENT SYSTEM
PENTABS
WORD MANAGEMENT SYSTEM
PENTABS
THE PENSION SALES CREATOR
PENTABS
PLAN DOCUMENT SYSTEM
PENTABS
THE CLIENT MANAGEMENT SYSTEM
PENTABS
401(K)/THRIFT OPTION
PENTABS
PARS
PHYSICIANS RESOURCE MANAGER-MO
EMS/PC
PROTOTYPE, INC.
MATCH MAKER+
QBIT COMPUTER SOLUTIONS
UPSTART
RAM SOFTWARE LIMITED
SYNERGY
RAM SOFTWARE LIMITED
FINANCIAL MANAGEMENT SYSTEM
RCP CONSULTING, LTD.
LAN:MAIL MONITOR
SOFTWARE CONNECTIONS
LAN:DATACORE
SOFTWARE CONNECTIONS
REAL WORLD
SOFTWARE SERVICES, INC.

DIRECT MAIL II
SOFTWARE SYNERGY GROUP
TABS III-M
SOFTWARE TECHNOLOGY, INC.
GENERAL LEDGER SYSTEM
SOFTWARE TECHNOLOGY, INC.
CRITICAL DATE SYSTEM
SOFTWARE TECHNOLOGY, INC.
TRUST ACCOUNTING SYSTEM
SOFTWARE TECHNOLOGY, INC.
EASE
STANHOPE ASSOCIATES, INC.
M.A.S. 90
STATE OF THE ART, INC.
SUMMIT BROADCAST SYSTEM
SUMMIT SOFTWARE SYSTEMS, INC.
SUND86A
SUNBELT COMPUTER SYSTEMS
SUNDB86M
SUNBELT COMPUTER SYSTEMS
MUNIS
THE COMPUTER CENTER
TENANTRAC
TIMBERLINE SOFTWARE CORPORATION
AEPEX
TIMBERLINE SOFTWARE CORPORATION
MEDALLION/SPECIALTY
TIMBERLINE SOFTWARE CORPORATION
MEDALLION/BUILDER
TIMBERLINE SOFTWARE CORPORATION

MEDALLION/GENERAL
TIMBERLINE SOFTWARE CORPORATION
MEDALLION COLLECTION ACCOUNTING
TIMBERLINE SOFTWARE CORPORATION
IMPULSE
TOM FAIREY COMPANY
TRACLINE
TRACLINE SOFTWARE, INC.
CUST. APPL: FINAN, MANU, DISTR
TRACY ASSOCIATES, INC.
KINDERGARTEN THRU 12TH COURSEWARE
WASATCH EDUCATION SYSTEMS
WORKERS' COMPENSATION CASE MGMT SYSTEM
WEISENBERGER AND ASSOCIATES
WELBER RM. & EVENT SCHEDULING
WELBER COMPANY
BUSINESS ACCOUNTING SYSTEM
WESTCON ASSOCIATES, INC.
OCS OFFICE CONTROL SYSTEM
WIBBEN CONSULTING & SYSTEM
LASER BEAM
WORDNET
DBXL
WORDTECH SYSTEMS, INC.
INFORMAL COMPLETE TRACKING SYSTEM
WUTC
RAILROAD INVENTORY
WUTC
PRODUCE BROKERAGE SYSTEM
ZADEK, INCORPORATED

PAYROLL
REALWORLD CORP.
ORDER ENTRY/BILLING
REALWORLD CORP.
SALES ANALYSIS
REALWORLD CORP.
ACCOUNTS RECEIVABLE
REALWORLD CORP.
MBSI ACCOUNTING
REALWORLD CORP.
RED RIVER FUEL PACKAGE
RED RIVER SOFTWARE
ESQL/C
RELATIONAL DATABASE SYSTEMS, INC.
C-ISAM
RELATIONAL DATABASE SYSTEMS, INC.
ROBEC SIMPLIFIED BUSINESS SYSTEM
ROBEC, INC.
UBS-2 UTILITIES BILLING SYSTEM
ROBITRON SOFTWARE RESEARCH
IPC-1 INVENTORY PRODUCTION
ROBITRON SOFTWARE RESEARCH
GLS-1 GENERAL LEDGER SYSTEM
ROBITRON SOFTWARE RESEARCH
FBS-1 FREIGHT BROKERAGE SYSTEM
ROBITRON SOFTWARE RESEARCH
ATS-1 AUTO TIRE STORE MANAGEMENT
ROBITRON SOFTWARE RESEARCH
APS-2 AUTO PARTS MANAGEMENT SYSTEM
ROBITRON SOFTWARE RESEARCH
AND MANY MORE . . .

for using NetWare.®

There are currently over 3,000 NetWare-compatible multiuser applications available. With more being introduced every day. No other local area network operating system can say the same. And that's reason enough for the 300,000 NetWare users throughout the world.

See for yourself.

Send for a copy of the new "NetWare-Compatible Software Directory" today. In it you'll find a comprehensive listing and descriptions of the thousands of applications available for NetWare-supported LANs.
Price: $50.

Yes, please send me ________copy(ies) of the Software Directory at $50 each. I'm enclosing a check, or charge my VISA or MasterCard (circle one).
Credit card # ______________________________ Expiration Date ________

Signature ______________________________

Name ______________________________

Title ______________________________

Company ______________________________

Address ______________________________

City ____________________ State ____________

Zip ____________________ Phone (______) ____________

NOVELL

Return to:
"Software Directory"
Novell, Inc.
748 North 1340 West
Orem, Utah 84057

MICROSOFT LANGUAGES NEWSLETTER Vol. 1, No. 8

News about the Microsoft Language Family

Improved Microsoft® C Compiler Version 4.00 Is Now Available

ANSI compatibility: Microsoft C Compiler Version 4.0 provides increased support for the developing ANSI standard. **Library support:** Over 30 new library functions have been added, including UNIX™ System V functions vprintf, vscanf, and setvbuf. **More memory models:** Compact and huge memory models have been added for increased versatility, bringing the number of supported memory models to five. **Optimizations:** Elimination of common sub-expressions joins our list of optimizations to make your generated code tighter than ever before. **Source code:** Full start-up and exit source code is provided. In addition to these new features, Microsoft C Compiler Version 4.0 allows you to compile and optimize larger programs, get mixed source and disassembly listings, control floating-point math better, and use improved overlay capabilities. And MAKE, the program maintenance utility, helps ensure up-to-date program components by automatically recompiling and relinking to keep up with your source code changes.

The New CodeView™ Source Debug Utility Is Bundled with Microsoft C. All the power of SYMDEB plus full windowing orientation make CodeView the debugger you will want to grow into. See source code and disassembly interspersed. Use drop-down menus, multiple windows, independent graphics screen swapping, and a read-only source code editor to navigate through your code and locate logic errors more quickly than ever before. Exercise your code with the dynamic C expression evaluator instead of recompiling multiple test cases. Set conditional breakpoints. See registers and variables (even locals) change as you watch an animated trace program execution. Special offer for CodeView demo: Words can't say it all here, but our freely copyable demo disk is available for only $5. Take a test drive!

New Microsoft QuickBASIC Version 2.00 Makes Programming in BASIC Easier and Faster

Microsoft QuickBASIC has undergone a major transformation in its new Version 2.0 release. We think that it will change the way you program in BASIC. The Microsoft QuickBASIC programming environment features an integrated editor, compiler, and runtime.

The editor uses a Microsoft Windows-like interface for full screen editing with pull-down menus. The editor supports the Microsoft Mouse and the IBM® 43-line Enhanced Graphics Adapter. The faster compiler can be invoked directly from the editor, compiling Microsoft QuickBASIC programs directly into memory and bypassing the link step. If no errors are detected the program can start executing immediately. Both compilation errors and runtime errors will restart the editor and place the cursor on the troublesome line.

Debugging compiled programs is easier with Microsoft QuickBASIC 2.0. The TRON statement enables the new source display animated trace and step modes. The tracing features can be toggled on and off using the function keys.

Large BASIC program development is supported by an additional new method of separate compilation. User libraries of BASIC and assembly language routines can be loaded into the in-memory compilation and execution environment. Microsoft QuickBASIC 2.0 also supports stand-alone application development.

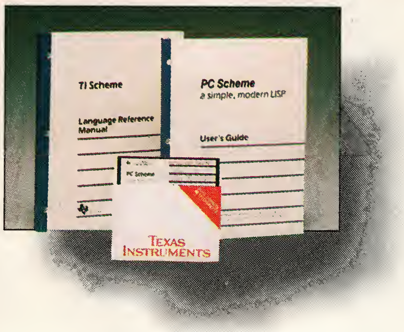
New Microsoft QuickBASIC language features include block IF/THEN/ELSE, Enhanced Graphics Adapter support including the new 640 by 350 graphics modes, and dynamic arrays that can now use all of available user memory.

For more information on the products and features discussed in the newsletter,
write to: Microsoft Languages Newsletter
16011 NE 36th Way, Box 97017, Redmond, WA 98073-9717
Or phone:
(800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638.

Latest DOS Versions:

Microsoft C Compiler	4.00
Microsoft COBOL	2.10
Microsoft FORTRAN	3.31
Microsoft Macro Assembler	4.00
Microsoft Pascal	3.31
Microsoft QuickBASIC	2.00

Look for the Microsoft Languages Newsletter every month in this publication.



PC Scheme

This fast LISP compiler runs on the base IBM PC hardware, yet meets professional programming standards.

Implementors of the LISP language have had a tacit agreement that the IBM PC hardware is not powerful enough to drive LISP software. And no wonder—the emerging standard for LISP, Common LISP, requires 10 times as much memory as a 640KB machine can provide. Memory and speed considerations relegate the PC and PC/XT to the lesser roles of training vehicles and target machines—or do they?

An examination of IBM PC LISP implementations confirms this dim view of what can be done on the PC. None is a complete implementation of a standard dialect; some are mere training tools; and one delivers, in the words of an author, "conceptual tinker toys for adults." Implementors provide some Common LISP functions, but fundamental differences between dialects prevent portability. Other factors mitigate against serious LISP development; designers have paid little attention to support environments or debugging. Most products are interpreters, so they are quite slow. (An exception is Golden Common LISP 286 Developer, a fast $1,195 compiler for PC/ATs with extended memory. For more information on Gold Hill's implementations, see "Creating a Standard LISP," Mark Bridger and John Frampton, December 1985, p. 98.)

Texas Instruments challenges conventional wisdom with PC Scheme, a full LISP development system tailored to the PC and compatibles. PC Scheme is complete, well-designed, and inexpensive at $95. For these reasons, *PC Tech Journal* has named PC Scheme Product of the Month for August.

Creating a full LISP system on a PC is a marvel somewhat akin to building a sailing ship in a bottle, so every step of PC Scheme's design was carefully focused on the task. Texas Instruments took the extra step of molding the language as well as the implementation to the microcomputer environment. PC Scheme is actually two products in one: the implementation and a language (called TI Scheme).

The TI Scheme language is different without being nonstandard. Scheme is a LISP dialect used for teaching at the college level. It was designed expressly for personal computers at MIT in 1975 and standardized in 1985. A number of books and papers have been published on Scheme, most notably, *The Structure and Interpretation of Computer Programs*, (see "An Elegant Scheme," Book Reviews, this issue, p. 183). TI Scheme was designed to run the examples in this book unmodified, a significant concession to the novice seeking to master this complicated language.

Block structure and simple syntax make Scheme the most elegant dialect in the LISP family. While respecting the language, TI Scheme's designers did not lose sight of the practical realities of programming. TI Scheme is a vast extension of the Scheme standard, which includes character and string handling, structures (similar to Pascal records), arrays, vectors, windows, graphics and interfaces to C and assembly language. The Scheme command SET! has the same generic replacement capabilities as Common LISP's SETF. The TI Scheme reference manual contains a description and example for each of the more than 400 functions.

The PC Scheme implementation is as tightly designed as the TI Scheme language specification. An incremental compiler processes all source code into virtual machine code, which is smaller but slower than native code. Slow or not, PC Scheme achieved the best overall times of any IBM PC LISP implementation on a function-calling and CONSing benchmark (TAK and DERIV in *Performance and Evaluation of LISP Systems*, MIT Press, 1986).

Attention to detail marks the PC Scheme implementation as much as its high speed. Such attention is especially important in LISP implementations because parenthesis-laden LISP code is not easy to read. PC Scheme's EMACS-style editor (the standard syntax-sensitive LISP editor) is spare but fast, providing a quick edit/compile cycle. Extras, such as session transcript, fast-loading files, and shell-to-DOS, are abundant. PC Scheme's smooth and bug-free operation contrasts markedly with other PC LISP implementations.

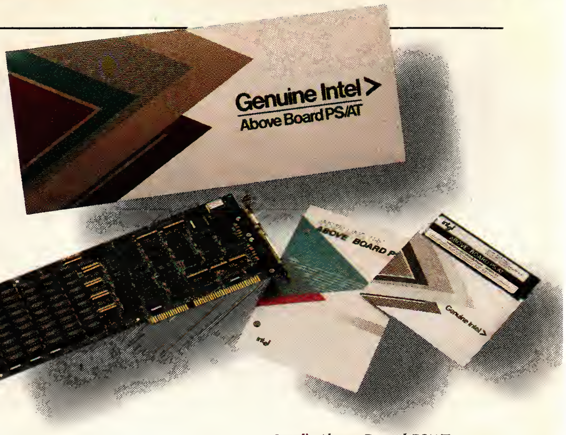
PC Scheme takes a unique and laudable approach to error handling. The TI Scheme language includes 22 error handling functions in addition to the normal TRACE command, providing all sorts of low-level hooks, breakpoints, and diagnostic print-outs. In addition, PC Scheme has another facility for debugging. While most LISP systems spawn error levels during interactive sessions, PC Scheme uses a single error level. When an error is encountered, the runtime debugger displays a command menu. The debugger displays the current lexical and dynamic state of the LISP system and allows the user to step through the system's nested blocks and environments. Scheme objects can be listed or modified from the debugger.

Texas Instruments has succeeded in an endeavor that most LISP vendors have not even tried. The PC Scheme environment is a complete LISP system that runs a standard dialect on the base PC hardware. Space limitations were considered during every phase of TI Scheme design and PC Scheme construction, resulting in a tool that is not a toy, not a compromise, but a complete LISP implementation for the professional programmer.

PC Scheme: $95
Texas Instruments, Inc.
Data Systems Group
P.O. Box 2909
M/S 2244
Austin, TX 78769-2909
800/527-3500
CIRCLE 343 ON READER SERVICE CARD

Hardware, software, and other developments for the IBM PC family

ShareData by Extended Systems



Intel's Above Board PS/AT

FROM IBM

IBM Corporation has announced that the National Service Division (NSD) now can remove or disconnect a non-IBM machine and/or non-IBM feature that is diagnosed as causing system or machine failure. Further, the NSD can replace the non-IBM machine or non-IBM feature with a similar non-IBM machine or non-IBM feature supplied by the customer. **Service support for non-IBM PC products** does not cover non-IBM machines or non-IBM features for which IBM offers maintenance agreement service. Service support for non-IBM products is available under an amendment to the IBM maintenance agreement. Annual charge of $30 per system unit (this is in addition to the maintenance agreement charge for the IBM PC system unit).

IBM BOOKS is a new department of the company that will publish books for business and computer professionals. The books will be sold in bookstores and other retail outlets; they will be written by IBM authors and other experts in the field of information processing. Distribution will be handled by Random House, Inc. (New York) to stores in the United States and Canada. Six titles are planned for publication this fall, covering topics such as portable computing, networking, business graphics, word processing, maintenance, and a dictionary of computer terms. IBM plans to publish about 20 books a year.

CAEDS for the **RT/PC** provides users with engineering capability for solid geometric modeling and finite element modeling. Using the data-transfer facilities of the PC 3278/79 Emulation Adapter and RT 3278/79 emulation, data can be passed among CAEDS products on an IBM host and CAEDS for the RT/PC on an RT. Data also can be passed between CAEDS for the RT and other IBM CAD/CAM products running on an IBM host. CAEDS base for the RT, $1,655; CAEDS object modeler for the RT, $6,060; CAEDS system modeler for the RT, $8,230; CAEDS graphics finite element modeler for the RT, $8,940.
IBM Corporation, 900 King Street, Rye Brook, NY 10573; Contact the local IBM dealer, 800/426-2468
CIRCLE 301 ON READER SERVICE CARD

HARDWARE

The **82786 graphics coprocessor** from **Intel Corporation** can manipulate windows in hardware 100 times faster than traditional software approaches. This high-performance, single IC supports two independent, on-chip processors for manipulating graphics and text while executing multiple windows. The 82786 operates independently of the host CPU; it is designed to work with all Intel microprocessors and requires minimal support circuitry for most system configurations. It uses the same advanced CHMOS-III process as the 80386 microprocessor. The 82786 coprocessor has 130,000 transistors, combines high performance and low power consumption, operates at less than 1 watt, and supports as many as 32 dynamic RAMs in its dedicated graphics memory. Less than $100 in quantities of 1,000.
Intel Corporation, Literature Department, W-300, 3065 Bowers Avenue, Santa Clara, CA 95051; 800/548-4725
CIRCLE 306 ON READER SERVICE CARD

Intel also has announced a combination expanded memory and multifunction board for the PC/AT. The **Above Board PS/AT** has a serial port, a parallel port, several standard software utilities, and up to 1.5MB of memory. With the Above Board piggyback option, memory can be boosted to 3.5MB. Above Board PS/AT memory can be divided, and used to fill in conventional memory (up to 640KB), as expanded memory consistent with the Lotus/Intel/Microsoft expanded memory specification, or as extended memory. 128KB version, $595; 512KB version, $695; Above Board/AT piggyback memory, $295 for a 128KB configuration.

A new 8-MHz version of the 80287 math coprocessor also has been introduced. The **80287-8** can boost numeric processing speeds of many Intel 80286-based accelerator boards. $410.
Intel Corporation, 5200 N.E. Elam Young Parkway, Hillsboro, OR 97124-6497; 800/538-3373; in Canada, 800/235-0444
CIRCLE 307 ON READER SERVICE CARD

The first of the new **ShareData** family of products designed to simplify data sharing has been announced by **Extended Systems**. These products allow up to four PCs to share concurrently as much data as they wish on one external 20MB hard-disk drive. Data security includes volume access protection and user access control. User log-on identification with or without password protection is available. An internal self-test diagnostic is activated each time ShareData is powered on and byte-by-byte error checking is provided for the interfaces between the computer and ShareData. The product can use RS-232 or RS-422 communication. The hard disk included with the ShareData package has a formatted capacity of 21MB with an average access time of 65 milliseconds. Nonrecoverable read errors are 1 per 10^{12} bits read. Other products offer sharing capability for printers (ShareSpool) and plotters (SharePlot). ShareData (ESI-3772) package, $1,995; high-performance ShareData (ESI-37729) package with four RS-422 interface cards, $2,395; RS-422 interface cards, $150 each, $545 for a bundle of four.
Extended Systems, P.O. Box 4937, 6062 Morris Hill Lane, Boise, ID 83711; 208/322-7163
CIRCLE 313 ON READER SERVICE CARD

PCI ControlLOGraph from Burr-Brown

By OmniTel, Inc.

Sigma Designs, Inc. has announced **FasTrak**, a portable, stand-alone streaming tape backup unit that allows both image and file-by-file backup and restore of data stored on PC hard disks. Two models are available. The **FTK-0020** has a built-in, half-height 5¼-inch tape drive that accepts one-eighth-inch cassettes with a storage capacity of 20MB or 27MB. The **FTK-0060** includes a half-height 5¼-inch tape drive, but it will use one-fourth-inch tape cartridges capable of storing 60MB of data. Both packages include Sigma's QIC 36 tape controller board, tape utility software, one tape cassette, and shielded cable. FTK-0020, $995; FTK-0060, $1,395.
Sigma Designs, Inc., 2023 O'Toole Avenue, San Jose, CA 95131; 408/435-1480
CIRCLE 302 ON READER SERVICE CARD

The **DNA Gateway** from **Network Development Corporation** is a 3270 communications link emulating a remote 3274 communications cluster controller. Standard DNA Gateway features include support for 32 concurrent SNA sessions, realtime multitasking execution, IBM 3274 model 51C/61C remote communications controller emulation, IBM 3178/3278 model 2 and 3178/3279-base two-and four-color support display emulation, IBM 3287 printer emulation, and the SNA Soberly Protocol Analyzer. 8-session configuration, $3,995; 16-session, $4,995; 32-session, $5,995.
Network Development Corporation, 81 Great Valley Parkway, Suite 700, Malvern, PA 19355; 215/296-7420
CIRCLE 304 ON READER SERVICE CARD

Omnipak is a single-board multifunction card integrated with a 1200/300-bps modem for use with the PC, PC/XT, PC/AT, and compatibles. Introduced by **Omnitel, Inc.**, the board features a Hayes-compatible modem that can be upgraded to 2400 bps, with 384KB of memory, both parallel and serial ports, a game port, and a clock/calendar with battery backup. Omnipak comes bundled with communications and utilities software. The modem included offers COM 1 through 4 addressability as well as call-progress reporting. $499.
Omnitel, Inc., 5415 Randall Place, Fremont, CA 94538; 415/490-2202
CIRCLE 310 ON READER SERVICE CARD

The **CMS** family of SCSI-compatible mass storage peripherals for the PC is available in internal or external configurations. Each subsystem consists of 20MB combination hard-disk and tape-backup peripherals, embedded VLSI controllers, and an SCSI host adapter. The external configuration, called the **SCSI Power Twin 20:20**, is packaged in a cabinet that fits alongside the CPU. A half-size PC host adapter card that

SCSI-compatible mass storage peripheral family from CMS

allows the PC to communicate with the SCSI peripheral bus is included; it can support both the disk and the tape drives in the Power Twin 20:20, as well as up to five other SCSI peripheral devices. The internal configuration includes a one-fourth-inch tape backup unit that can replace an internal floppy-disk drive and CMS's 20MB SCSI Drive Plus expansion card, which includes the half-card host and a 3½-inch disk drive. External configuration, $1,950; internal, $1,660; internal tape drive and host adapter card, $910; Drive Plus, $990.
CMS, 401-B W. Dyer Road, Santa Ana, CA 92707; 714/549-9111
CIRCLE 308 ON READER SERVICE CARD

The **PCI ControLOGraph System** from **Burr-Brown** is a fully integrated system of hardware and software for data logging, graphics display, alarm annunciation, and digital control. Data input choices include 21 analog signals, 24 digital bits, and 3 frequency/event counting channels. Four ranges of input voltage as well as J, K, or T thermocouples can be selected via menu. Analog, digital, and frequency/event data all are acquired on each data acquisition scan performed by the ControLOGraph. The data can be displayed in realtime, as they are being stored on disk. The system's software can restart itself in the event of a power failure; after power is restored, ControLOGraph automatically reboots the computer. Hardware/software system, $2,750; software, $795.
Burr-Brown, International Airport Industrial Park, 6730 S. Tucson Blvd., P.O. Box 11400, Tucson, AZ 85734; 602/746-1111
CIRCLE 309 ON READER SERVICE CARD

The **QMS-PS 800** from **Quality Micro Systems** combines the technology of a laser printer with that of the PostScript typesetting language from Adobe Systems, Inc. This printer (which prints eight pages per minute) enables the user to include text, line art, halftones, textures, grayscaling, patterns, and images in any size with a resolution of 90,000 dots per square inch. $5,895.
Quality Micro Systems, P.O. Box 81250, Mobile, AL 36689; 205/633-4300
CIRCLE 318 ON READER SERVICE CARD

Through a joint OEM agreement, **Computer Identics Corporation** will handle the national distribution of **Micro Peripherals, Inc.'s** two bar code printers. The **Printstar 200** and the **Printstar 345** (the numbers refer to the length of the printable line in millimeters) produce optically scannable output in the form of bar codes, as well as machine-readable OCR-A. The two


Tseng Labs' EVA (Enhanced Video Adapter)


T4 Color Graphics Controller by Microfield Graphics

printers are enhanced versions of the MPI BarMate 9 and BarMate 15. All BarMate printers have Code 39, Code 128, and Rationalized Codabar symbologies suitable for LOGMARS, AIAG, and HIBC as standard. The programs for the bar code and OCR-A symbologies are resident in the PROMs of the printers and produce the symbologies independent of special software. In IBM-compatible mode, these printers handle all printing requirements of a PC. Friction and tractor feed allow for many types of paper, including labels and continuous form. Printstar 200, AI-66501-1, $1,395; Printstar 345, AI-66502-1, $1,995; AI-66502-2 (with RS-232 interface), $2,105.
Computer Identics Corporation, 5 Shawmut Road, Canton, MA 02021; 617/821-0830
CIRCLE 311 ON READER SERVICE CARD
Micro Peripherals, Inc., 4415 S. 500 West, Salt Lake City, UT 84123; 801/263-6000
CIRCLE 312 ON READER SERVICE CARD

ITT Information Systems has announced the **ITT XTRA XL**, a high-performance, 80286-based microcomputer that runs DOS and XENIX. The system features 8-MHz frequency, zero-wait-state memory, dynamic disk I/O caching, an average hard disk access time of 28 milliseconds, and an optional 80287 coprocessor. An 8-MHz 80186-based communications coprocessor offers speed in multiuser configurations. The ITT XTRA XL can be used as a LAN server (models I and II) or as a multiuser system (models III and IV). All models include a battery-backed realtime clock, an audio speaker, two serial ports and one parallel port on the motherboard, six PC/AT-compatible expansion slots, and three PC/XT-compatible expansion slots. Prices begin at $5,299.
ITT Information Systems, 2350 Qume Drive, San Jose, CA 95131; 408/945-8950
CIRCLE 305 ON READER SERVICE CARD

From **Tseng Laboratories, Inc.** comes **EVA**, an IBM EGA-compatible enhanced video adapter that is equipped with a parallel port. It uses the proprietary ET2000 VLSI Graphics Chip Set designed by Tseng Labs and is intended as a superset of the EGA. Features of EVA include a hardware zoom, a viewport/zoom window, a turbo graphics engine, a 132-column screen, and light pen pixel manipulation. EVA drives monochrome, color, and enhanced color displays and offers 256KB of display memory, 16 user-selectable colors from a color palette of 64, a split screen, smooth scrolling and pixel panning, a soft downloadable font capability, and an 80-by-25/43 text display. $525.

Optional hardware compatibility for the Hercules, CGA, and monochrome adapters (beyond the BIOS level compatibility standard with EVA) is achieved with the optional Compatibility Module II daughterboard. $50.
Tseng Laboratories, Inc., 205 Pheasant Run, Newtown, PA 18940; 215/968-0502
CIRCLE 320 ON READER SERVICE CARD

Microfield Graphics, Inc. has announced the **T4 Color Graphics Controller** for the RT. The T4 is a 1,024-by-800, four-plane color controller that uses a VLSI-based bit-slice processor to provide high-speed, high-resolution graphics. It is supported by an ANSI CGI interface permitting its use on either the RT's RISC processor or its optional DOS coprocessor. The CGI interface permits applications to access the line drawing, polygon and text manipulations, and window-management microcode of the T4 directly from C subroutines, while incurring minimal overhead on the RISC or DOS processors. $3,200; RT support software, $150.
Microfield Graphics, Inc., 8285 S.W. Nimbus Avenue, Suite 161, Beaverton, OR 97005; 503/626-9393
CIRCLE 317 ON READER SERVICE CARD

The **GhostWriter** is a new multiformat software copier product from **Magnetic Designs** that allows automatic duplication of all common brands (formats) of diskette software. It includes a plug-in card for the IBM PC and the SmartCopy software package, which allows even first-time users to duplicate and validate software easily. Key features include automatic analysis and duplication of diskettes, the ability to duplicate as many as 80 IBM PC or 163 Commodore diskettes per hour with an optional diskette autoloader (eliminates reloading), the ability to store multiple software masters on the system hard disk, support of direct connection to floppy-disk drives, and variable bit-cell timing with precompensation. Optional extensive analysis tools are available. $1,995.
Magnetic Designs, 643 Bair Island Road, Redwood City, CA 94063; 415/363-1141
CIRCLE 315 ON READER SERVICE CARD

A high-speed, high-capacity optical disk system for users of IBM PC-based CAD/CAM systems has been introduced by **N/Hance Systems**. The **N/Hance 525** system provides 230MB of formatted storage and includes the optical disk drive, a 5¼-inch disk, an IBM PC short-board controller, and device driver software. The N/Hance 525 is a WORM (write-once-read-many-times) system that allows users to write data to the disk, store them in a secure, unalterable form, and read them quickly when needed. The system has a transfer rate of 2.5 Mbps, a track-to-track access time of less than 1 millisecond, and an average access time comparable to that of a Winchester disk. $3,495. Also available is a text database management system that allows users to search the database by keyword or phrase, priced at $1,000.
N/Hance Systems, 908R Providence Highway, Dedham, MA 02026; 617/461-1970
CIRCLE 316 ON READER SERVICE CARD

READ  ONLY

A review of the IBM Personal Computer Family. Vol. 3, No. 1

Welcome To *Read Only.*

Here's great news for IBM PC users. IBM has expanded its already expansive PC product line to bring even more power and flexibility to your desktop.

In this issue alone you'll be reading about new enhancements to the IBM PC/XT, the IBM Personal Computer AT,® the IBM PC Keyboard, the IBM Proprinter, the Mainframe Communications Assistant, and more.

And, as if that weren't enough, IBM has also found time to expand your computing horizons even more by introducing exciting new PC products. This issue of *Read Only* will tell you about the new IBM PC Convertible, new 3.5″ diskettes, the new IBM PC 3.5″ External Diskette Drive, and two new accounting software packages.

HARDWARE NEWS

The Right Touch.

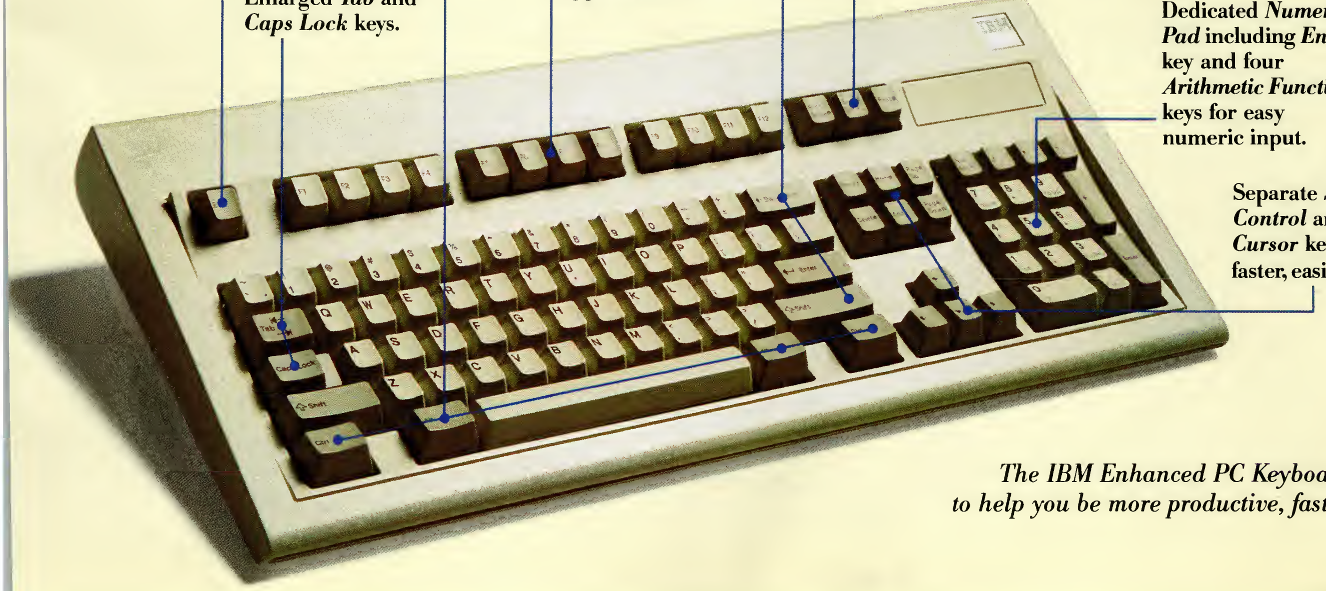
To make it easier than ever to work at an IBM PC/XT or Personal Computer AT, IBM has introduced the new IBM Enhanced Personal Computer Keyboard (shown below).

IBM redesigned its classic keyboard to better meet the needs of PC users, office system users, as well as users who communicate with larger computers.

To accomplish this, IBM included separate cursor and screen control keys, making it easier for users to dedicate the numeric keypad to numeric input when working with number-intense applications. Plus, the keypad can still be used for cursor and screen control when not in the num/lock mode.

IBM also increased the number of function keys from ten to twelve, arranged across the top of the keyboard. This gives users two additional keys for increased automatic operation.

All to enhance the productivity of those who work with words, numbers or a host of other applications.



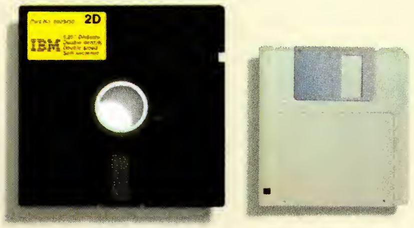
The IBM Enhanced PC Keyboard to help you be more productive, faster.

Small Wonder.

Proof positive that good things come in small packages is the new 3.5-inch diskette.

And that's big news for anyone who uses an appropriately configured IBM PC, PC/XT, Personal Computer AT and the new IBM PC Convertible.

Durable 3.5″ diskettes, allow you to carry over 350 standard typed pages (720KB) in your shirt pocket. That's twice as much information as on a 5.25″ (360KB) diskette, in a much more rugged, portable form.

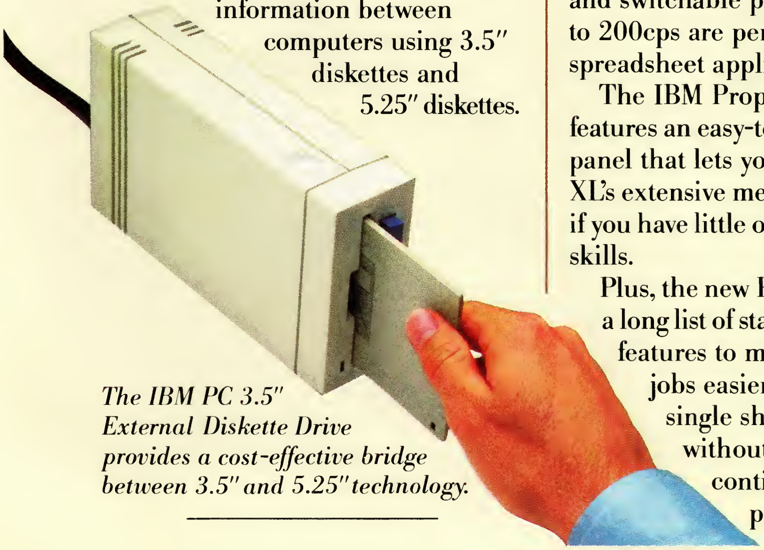


The new 3.5″ diskette stores up to twice as much (720 KB) as a 5.25″ (360KB) diskette.

Now IBM PC users have the option of using programs and data in either size, and the flexibility to work with other members of the IBM PC family using 3.5″ diskettes.

Translation Services.

The new IBM Personal Computer 3.5″ External Diskette Drive provides a vital bridge between your IBM PC and 3.5″ technology. This compact unit makes it easy for you to share information between computers using 3.5″ diskettes and 5.25″ diskettes.



The IBM PC 3.5″ External Diskette Drive provides a cost-effective bridge between 3.5″ and 5.25″ technology.

The new IBM PC Convertible is a j

The IBM Personal Computer 3.5″ External Diskette Drive comes in two models, one for the IBM Personal Computer AT and one for the IBM PC, PC/XT or IBM PC Convertible.

Information and applications can be shared* between 3.5″ diskette drive machines and an IBM PC running DOS 3.2 with the IBM Personal Computer 3.5″ External Diskette Drive attached.

Transferring files and programs between 3.5″ and 5.25″ diskettes is as easy as making a backup copy of a diskette. So, you can very quickly have a "database to go" for your IBM PC Convertible. Or, a week's worth of sales call information in a ready-to-use form for your secretary when you return.

Print Evolution.

The new IBM Proprinter XL is especially designed to make life easier for those who work with accounting applications. Its wide carriage design and switchable printing speeds of up to 200cps are perfectly suited for spreadsheet applications.

The IBM Proprinter XL also features an easy-to-use front operator panel that lets you choose from the XL's extensive menu of features, even if you have little or no programming skills.

Plus, the new Proprinter XL offers a long list of standard, labor-saving features to make many printing jobs easier: easy printing of single sheets and envelopes without removing your continuous forms paper, power-assisted paper loading, all-points-addressable graphics capabilities, near-letter-quality printing (40cps), and emphasized text printing (100cps). Plus, you can set the printer in double high, double wide or emphasized print through the operator panel or through software.

Power To Go.

The new IBM PC Convertible can play two powerful roles in any business-person's life.

In the office, with an optional IBM PC monochrome or color display and adapter, the PC Convertible fills the bill as a space-saving desktop PC.

But when you're ready to hit the road or runway, just attach the high quality, 80-column x 25-line detachable LCD display, and the PC Convertible is ready to travel, too.

Weighing in at a scant 12 pounds, the IBM PC Convertible delivers full-size PC performance in a portable computer with heavyweight features including:

A full-function keyboard with full-size keys and the same center-to-center key spacing as a standard IBM PC keyboard.

A fast, very efficient 80C88 micro-processor with up to 10 hours of non-stop computing power between battery recharges (with average use).

Up to 512KB of user storage (through 128KB expansion cards from a standard 256KB).

Dual 3.5″ diskette drives supporting 720KB capacity 3.5″ diskettes.

Additional IBM PC Convertible features help ensure that work done on the road doesn't get lost in transit

...wer PC that works wherever you do.

These optional features include:

An internal modem feature to let you communicate with other computers simply by plugging the PC Convertible into any standard modular phone outlet.

The IBM PC Convertible Printer for system battery powered, near-letter-quality printing anywhere.

Plus, the IBM PC Convertible can come with a helpful set of programs to get you up and running on the road to enhanced productivity. Fast.

Planned For Growth.

It's a classic case of a very good thing that just keeps getting better.

IBM has now introduced the *enhanced* PC/XT product family: increased flexibility in storage, memory and option configuration for maximum productivity today, and

The new IBM PC/XT—enhanced power and flexibility for today. And tomorrow.

plenty of room to expand as your business does.

The XT is now available with a 20-megabyte hard file that can store up to 10,000 pages of information. There's also an easy, low cost way to increase memory to a full 640KB on the system board without tying up valuable expansion slots. And the XT now has 3.5″ diskette capability, utilizing the new IBM Personal Computer 3.5″ External Diskette Drive.

Plus, full IBM PC compatibility means that no matter which XT model you choose, you can benefit from the extensive IBM Personal Computer software library.

The new IBM PC/XTs, because one size should not have to fit all.

Power Play.

If you thought you'd seen all the IBM Personal Computer AT has to offer, think again. Twice.

IBM has increased processor speed in two new models of the IBM Personal Computer AT by an impressive 33% (from 6mhz to 8mhz). So, they're sure to become your fast friends if you work with large spreadsheets and volumes of data.

The new models offer a standard 30MB hard file and the option to add an additional 20MB or 30MB hard file. That's a grand total of 60MB, or approximately *30,000* pages of words and numbers.

These two newcomers are born communicators: Sharing files with other PC s from a variety of popular software programs. Working as a server for data storage and file processing in an IBM PC LAN running the IBM PC Local Area Network program. Utilizing IBM TopView™ and one of the IBM PC 3270 Emulation Programs to access mainframe information and to execute PC DOS and mainframe applications concurrently.

And the best news of all is that you can get all this increased power without an increased price.

Now the IBM Personal Computer AT offers a choice of hard files to meet your storage needs.

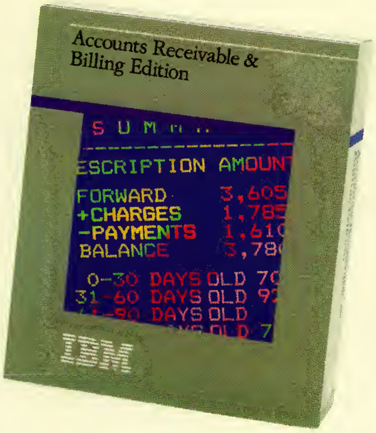
WHAT'S THE PROGRAM

Meet Your New Assistant.

Keeping the books for a small business is a very big job. So, IBM thought you could use a friendly, versatile, highly skilled assistant. The new IBM Accounting Assistant Series.

This complete series of automated bookkeeping application programs for small to medium size businesses can help cut any job down to size. And, its modular design means that you can start out with just the programs you need today, then expand your series as your business grows.

The IBM Accounting Assistant Series includes six individual editions: General Accounting, Accounts Payable, Accounts Receivable and Billing, Payroll, Inventory



The IBM Accounting Assistant Series. Big help for small to medium size businesses.

Control and Purchasing, and Job Cost bidding software.

Plus, thanks to the IBM Accounting Assistant Series' user-friendly attributes and easy-to-follow instructions, you can start profiting from your system from day one.

High-Powered Advice.

Perhaps your business has progressed past the need for basic accounting software. Then you should consider getting powerful, sophisticated help: the IBM Business Advisor.

IBM Business Advisor PC accounting software takes integrated software to a new level of sophistication and ease-of-use.

Business Advisor's seamless architecture allows functions from each of its modules to play together. Passage back and forth among the General Accounting, Accounts Payable, Accounts Receivable, Payroll, Inventory Control, Order Entry and other applications is intuitive. Menus guide you easily from one task to another.

And when you make a change anywhere, consider it made everywhere it applies. Automatically, through Business Advisor's real-time posting feature.

This Business Advisor speaks your language, too. It uses business language instead of accounting language, and has over 80 easily customized financial report formats built in. And you don't have to keep this all to yourself, because in an IBM LAN different people at different PC s can work on the same file at the same time.

And you always go further when everyone is working together.

"NEWS" BRIEFS

The 3270 Emulation Programs, Entry Level, Version 2 and Version 3, give you an easy and inexpensive way to attach your IBM PC to your host computer.

Working at your IBM PC, stand-alone or in an IBM LAN, you can now utilize the local power and user-friendly attributes of your PC for DOS applications, plus have access to the vast memory, number-crunching capacities and other productivity-enhancing capabilities of your host computer.

The Entry Level product offers up to 40% faster file transfer between your PC and host computer. A "Hot Key" for easy switching between host and PC applications. Keyboard remapping so you can always work in a familiar keyboard format. And much more.

Versions 2 and 3 can provide an economical gateway which lets you share the wealth of host knowledge with the members of your IBM PC Network or Token-Ring Network.

IBM PC 3270 Emulation Products can put your IBM PC or IBM LAN in touch with the big time.

Versions 2 and 3 support the new PC DOS 3.2, 3.5″ media, the IBM Local Area Network program 1.10, a host of printers and the TopView 1.1 interface, for multitasking and windowing capabilities.

All of which adds up to added productivity for you and everyone on your IBM Local Area Network or token ring.

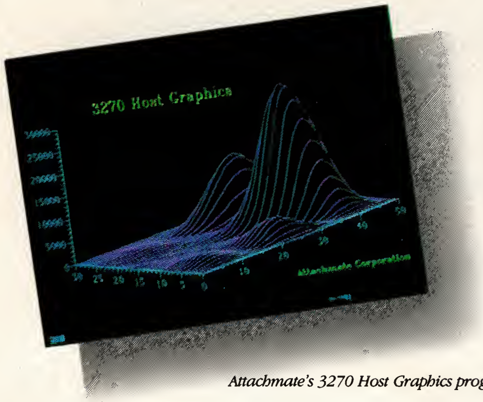
To find out more about the entire family of 3270 PC Emulation Programs, as well as a wide range of other IBM connectivity hardware and software, watch for the next *Read Only.*

Mainframe Communications Assistant enhancements include 3.5″ media support and increased IBM PC family communications capabilities.

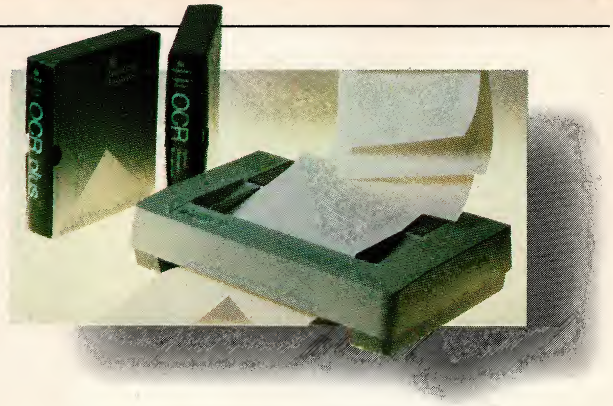
The IBM PC Voice Communications Option speaks for itself. This multifunction adapter card can allow your IBM PC to recognize and respond to voice commands, speak text that appears on the screen, initiate and receive/record/playback phone calls, provide remote, tone push button phone access to your PC (and host), and transmit voice and data simultaneously.

IBM®

For more information on any of the Personal Computer products discussed in this issue of *Read Only*, see your Authorized IBM Personal Computer Dealer. Or, call **800-447-4700.** In Alaska call **800-447-0890.**



Attachmate's 3270 Host Graphics program



JetReader Plus by Datacopy Corporation

A high-performance scanning system that provides OCR (optical character recognition) and image processing capabilities has been announced by **Datacopy Corporation**. The **JetReader** is a compact, lightweight image scanner with high resolution and an automatic paper feed. It comes with an interface to the IBM family of PCs as well as Datacopy's OCR and OCR Plus software and Word Image Processing System software. JetReader accepts a stack of as many as 10 sheets of legal-size paper for automatic feeding, and transfers them to an output tray in proper order after scanning. The image scanning is based on charge-coupled-device technology and provides selectable resolutions of 300 or 200 dots per inch (dpi). Scan times are approximately 43 seconds at 300 dpi and 28 seconds at 200 dpi. For OCR, the conversion time depends on the quality of both the original document and the computer in use; full page conversions typically take less than a minute for scanning and translation. Also available from Datacopy is another scanner, the **JetReader Plus**, which includes OCR Plus software for user training to recognize unknown type styles. JetReader, $2,950; JetReader Plus $3,250.
Datacopy Corporation, 1215 Terra Bella Avenue, Mountain View, CA 94043-1834; 415/965-7900

CIRCLE 314 ON READER SERVICE CARD

The **NMS R8000A** is a controller from **National Memory Systems** designed for high-performance, high-capacity disk/tape memory systems. This controller offers support for two SMD drives along with up to 4 SCSI interface devices on a single board for PC/ATs and RTs supporting UNIX and DOS applications. The board features switch-selectable DMA channels and port address to make installation in networks, varied CPUs, or large systems environments conflict-free. The R8000A is compatible with all major networks; it can simultaneously support the NMS PC.25, a one-fourth-inch cartridge tape system, and the NMS 007 (a 1,000MB laser optical cartridge disk that uses 12-inch media). $895 in OEM quantities.
National Memory Systems, 355 Earhart Way, Livermore, CA 94550; 415/443-1669

CIRCLE 303 ON READER SERVICE CARD

SOFTWARE

The **3270 Host Graphics Program** from **Attachmate Corporation** is a software package that lets users access host-generated color graphics. The product uses Attachmate's IBM IRMA-compatible 3-N-1 Coax Adapter and 3270-PC emulation program to manage four concurrent host sessions. Any combination of 3270 text or graphics sessions as well as a PC application can be active simultaneously. Attachmate uses the PC's all-points-addressable graphics and a unique algorithm to match host graphics screens to the resolution of the CGA or EGA on the PC. Thus, it can display full 3270 graphics screens without special adapters. Host graphics screens can be saved to disk, then edited locally using programs such as Z-Soft's PC Paintbrush. Output from the 3270 Host Graphics Program also can be routed directly to a PC printer, including IBM's 3852 Color Jetprinter. $595; 3-N-1 Coax Adapter, $1,195.
Attachmate Corporation, 3241 118th SE, Bellevue, WA 98005; 206/644-4010

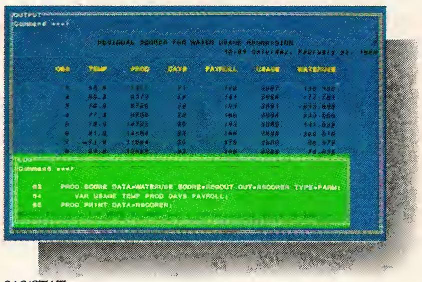
CIRCLE 324 ON READER SERVICE CARD

Release 3.0 of **BABY/36** is now available from **California Software Products, Inc.** BABY/36 lets a PC user develop, edit, compile, and execute new and existing RPG II programs; release 3.0 includes enhancements in step with version 3.0 of IBM's System Support Procedures for the System/36. REDE, READP, and *LIKEDEFN operation codes have been added. Noncontiguous index keys are now supported in the RPG compiler and BLDINDEX operation code. Maximum key length is increased to 63 bytes. MOVEA has been enhanced to allow numeric fields, and EVALUATE has been enhanced to support multiplication and division. Other features have been added for creating and maintaining libraries, printing data files, and editing and compiling programs and screens in an on-line environment. Prices range from $700 for the runtime version to $3,500 for a software development system.
California Software Products, Inc., 525 N. Cabrillo Park Drive, Santa Ana, CA 92701; 714/973-0440

CIRCLE 335 ON READER SERVICE CARD

A full-function statistical package for the PC, PC/XT, and PC/AT has been released by **SAS Institute Inc.** The **SAS/STAT** software performs both simple and complex analyses. The package includes interactive procedures for regression analysis, analysis of variance, common factor and component analyses, discriminant analyses, and scoring. To use SAS/STAT, sites first need PC-based SAS software, which includes tools for data management, descriptive stastistical analysis, and report writing. First-year license for corporate customers with up to 50 workstations, $2,000.
SAS Institute Inc., P. O. Box 8000, SAS Circle, Cary, NC 27511-8000; 919/467-8000

CIRCLE 325 ON READER SERVICE CARD



SAS/STAT screen

Ada compiler for the PC/AT by Alsys

EE Designer integrated software from Visionics Corporation

From **Visionics Corporation** comes **EE Designer**, a CAE/CAD integrated software package for electronic circuit design, layout, and fabrication applications. EE Designer consists of three integrated programs for schematic design, circuit simulation, and printed circuit board layout. It provides for automatic packaging of the engineer's design schematic into functional blocks that can be added to the system library of components consisting of standard TTL parts. Features of the package's schematic design module include auto-placement and back-annotation. The circuit simulation package includes a library of 150 common TTL and CMOS devices, including gates, flip-flops, counters, buffers, decoders, and RAMs. The user can expand the library by creating and adding unique shapes and components. The printed circuit board layout module of the package provides all functions necessary to design a printed circuit board as well as to produce artwork masters for fabrication, pen plotter, and photo plotter outputs. $975.
Visionics Corporation, 1284 Geneva Drive, Sunnyvale, CA 94089; 408/745-1551
CIRCLE 323 ON READER SERVICE CARD

Smalltalk-AT is the complete Smalltalk-80 version 2 for the PC/AT. Smalltalk-80 is an object-oriented programming environment previously available only on large, dedicated machines. Released by **Softsmarts, Inc.**, Smalltalk-AT includes the Xerox Smalltalk-80 source code, the Xerox image, and Softsmarts' ST-80 virtual machine, which is a true multiprocessing system. Smalltalk-AT features bit-mapped graphics, a window-based interface, all of the Smalltalk-80 version 2 classes, system browsers, a compiler, a profiler, and an interactive debugger that supports realtime debugging. $995.
Softsmarts, Inc., 4 Skyline Drive, Woodside, CA 94062; 415/327-8100
CIRCLE 339 ON READER SERVICE CARD

Alsys, Inc. has announced that its **Ada compiler** for the PC/AT has been validated through the Ada Validation Facility of the Ada Joint Program Office of the United States Department of Defense. A key feature of the Alsys Ada compiler is its use of protected (virtual) mode to overcome the 640KB limitation that is imposed by DOS. This permits access to extended memory up to 16MB and the accommodation of serious Ada programs on a PC. The compiler is bundled with a 4MB memory board. $2,995.
Alsys, Inc., 1432 Main Street, Waltham, MA 02154; 617/890-0030
CIRCLE 337 ON READER SERVICE CARD

Tominy, Inc. has announced **MACH 1 Professional Application Developer** (PAD), a single-user system generation facility that provides comprehensive applications development facilities for software practitioners. Product components include a relational/network/hierarchical DBMS, a fourth-generation programming language called LOGIC, a powerful applications program generator, an automatic screen editor, an efficient report generator, comprehensive system utilities, and flexible query. $199.95.
Tominy, Inc., 4221 Malsbary Road, Cincinnati, OH 45242; 800/543-8628; in Ohio, 800/445-1737
CIRCLE 338 ON READER SERVICE CARD

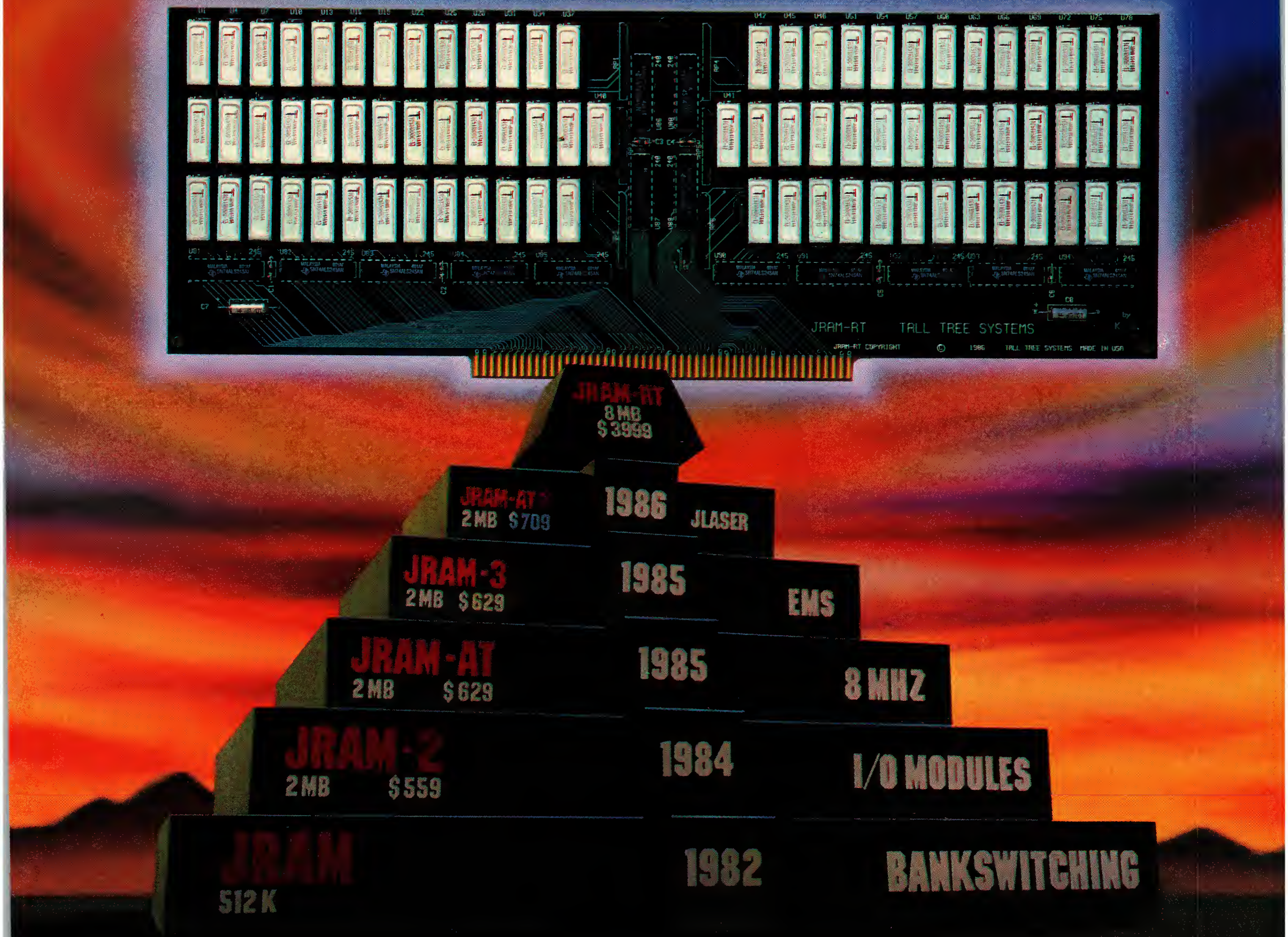
Three new products have been added to **LOGITECH, Inc.**'s family of Modula-2/86 development tools. A Turbo Pascal-to-Modula-2 translator called the **Turbo Translator** reads an existing Turbo Pascal program and generates a Modula-2 translation. It comes bundled with several new Modula-2 library modules, including screen-handling, graphics, and file-handling modules, which have been adapted to correspond directly to Turbo Pascal functions. $49.95.

The **Make Utility** runs through a program after changes have been made to the source file and selects modules affected by the changes. The utility generates a batch file containing a list of the compilation required to correct the system, providing the user with a map of all the files it has analyzed, as well as a list of which files need to be recompiled and why. It identifies new dependencies and gives a cross reference of exported/imported identifiers. $29.

A third new tool is the **Windowing Package**, which enables users to build windowing into their Modula-2 code. The package features virtual screens, color support, and overlapping windows with a variety of borders. It uses only 15KB of code. $49.
LOGITECH, Inc., 805 Veterans Blvd., Redwood City, CA 94063; 415/365-9852
CIRCLE 332 ON READER SERVICE CARD

TPORT-05, a systems-level program that enables the porting of mainframe- and minicomputer-based graphics software to the PC has been announced by **Grafpoint**. The new package resides under DOS as a device driver, allowing PC applications programs to use Tektronix 4105 graphics commands. The package processes Tektronix-like commands and escape sequences, thereby eliminating the need to rewrite software to conform to the virtual device interface standard for the PC. $495.

Also from Grafpoint comes **TNET-05**, a high-performance Tektronix 4105 emulation package running on PCs configured in a LAN. TNET-05, a device driver that processes Tektronix 4105 commands under DOS, gives network users the display and processing capabilities of a Tektronix graphics terminal on a PC. The package includes software that connects with communications software packages from Bridge Communications (EtherTerm) and Network Research Corporation (Fusion). $495.
Grafpoint, 4340 Stevens Creek Blvd., San Jose, CA 95129-1102; 408/249-7951
CIRCLE 328 ON READER SERVICE CARD



TALL TREE SYSTEMS.
A Technological Innovator.
Always a Step Ahead!

For true industry leadership, look no further than Tall Tree Systems.

We have a history of being first.

We were the first to introduce bankswitching. The first with two megabyte memory boards. The first with I/O modularity in a single slot. The first with 8 MHz speed capabilities. The only maker of single command EMS boards. The first with a laser printer solution — JLASER — that allows you to do full-page graphics and multiple type fonts on any Canon® or Ricoh® laser engine.

Now, we're first again with memory expansion for the IBM®RT.

Innovation is our tradition.

Our trademark is superior technology at the lowest possible price.

TALL TREE SYSTEMS

1120 San Antonio Road • Palo Alto, CA 94303 • (415) 964-1980

CIRCLE NO. 197 ON READER SERVICE CARD

PASCMAN screen

By Wiley Professional Software

NETWORK-OS is a 16-bit, NetBIOS-compatible network operating system that supports DOS 3.1 or later, all major network topologies, and Novell file and record locking. Introduced by **CBIS, Inc.**, NETWORK-OS is menu driven and employs a unique visual network shell (VNS) user interface. Commands are presented on hierarchical, pull-down screens. LAN resources are addressed by user-defined object names and mapped by mouse or keyboard. Four users per PC/XT or 16 per PC/AT are supported. When more workstations are required, multiple file servers may be employed.

In addition, CBIS announced a new open-bus twisted-pair wire interface able to transmit data at 2.5Mbps. This is three times faster than the existing RS-422 interface (which runs at 800Kbps). NETWORK-OS, $995; interface boards, RS-422, $295; twisted-pair, $395.
CBIS, Inc., 2323 Chesire Bridge Road, Atlanta, GA 30324; 404/634-3079
CIRCLE 330 ON READER SERVICE CARD

An advanced software tool called **CUBIT** has been announced by **SoftLogic Solutions, Inc.** This utility enables users to reduce the number of bytes required to store a file. CUBIT, comprised of four programs, is designed to compress and decompress data files. CUBIT's internal dictionary performs a look-up operation on each word in a test file to determine if the word is included in the dictionary. If it is, the word is reduced to one or two bytes of encoded data. For words that are not in the dictionary, CUBIT uses two additional techniques. One assigns new, shorter codes to the words, the other analyzes the frequency of character strings in the data. All programming is transparent and automatic. CUBIT also compresses binary files by adapting its algorithm. $49.95.
SoftLogic Solutions, Inc., 530 Chestnut Street, Manchester, NH 03101; 603/627-9900
CIRCLE 329 ON READER SERVICE CARD

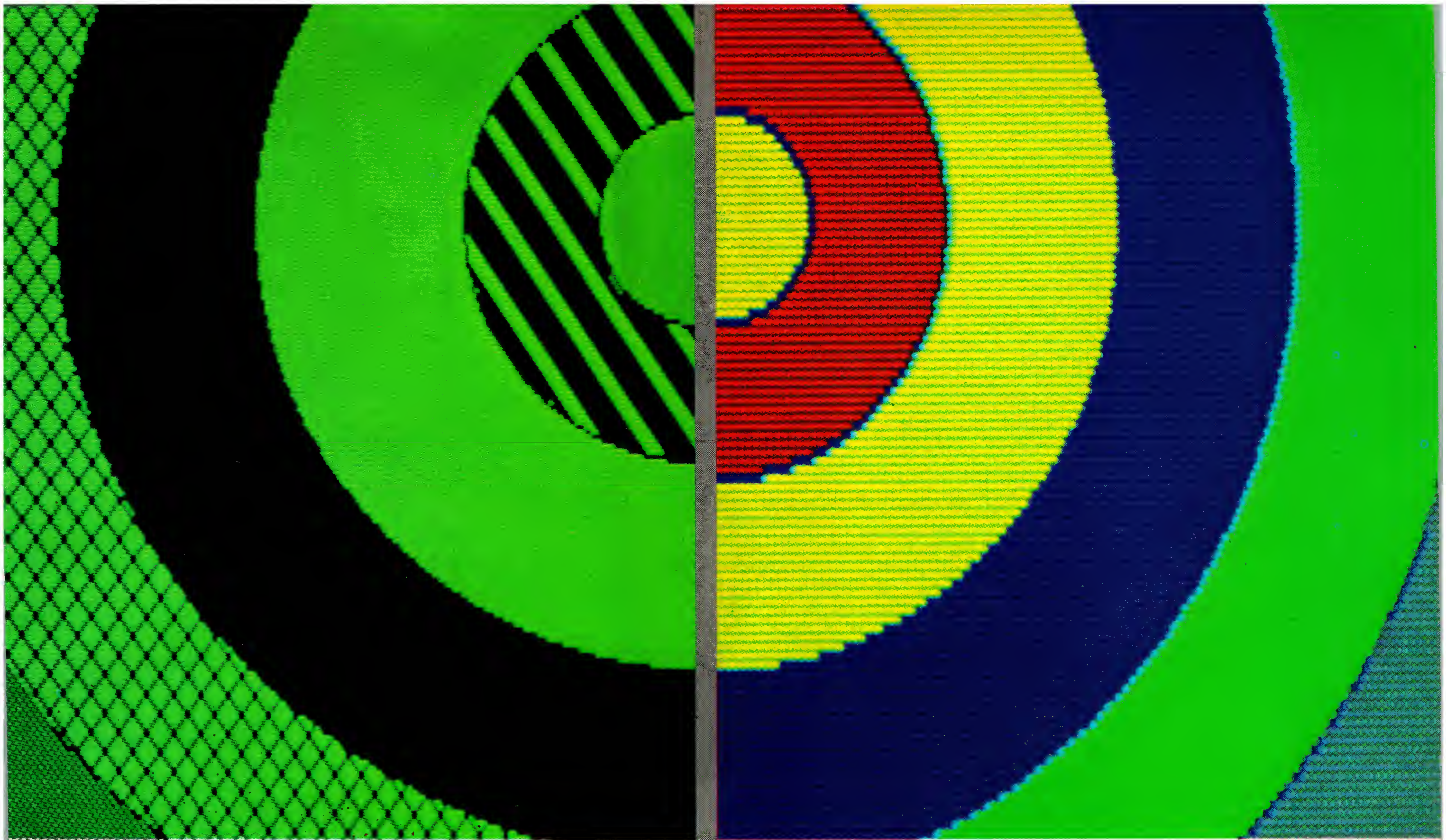
A programming tool for users of the IBM data acquisition and control adapter has been introduced by **Wiley Professional Software**. Called **LabSoft**, this program, which was originally developed by Cyborg Corporation for use with its ISAAC series of computer-based data acquisition and control systems, can be used to acquire, manipulate, and analyze real-world data. LabSoft is available in three versions—FORTRAN, BASIC, and C—to provide complete language compatibility; it features more than 40 programming tools/commands. FORTRAN or BASIC version, $350; C supplement, $75.
Wiley Professional Software, 605 Third Avenue, New York, NY 10158; 212/850-6788
CIRCLE 342 ON READER SERVICE CARD

Financial Software Systems, Inc. has released **PASCMAN** (Pascal Screen Management System) for use with Borland International's Turbo Pascal. With PASCMAN, a programmer can design colorful computer screen displays quickly. Using PASCMAN's full-screen editor, the user can set up screens identifying input and output fields. The editor handles all screen design, including colors, graphic characters, and an easy box drawing facility. Help screens and pop-up windows for every variable field on a screen can be created. $29.95.
Financial Software Systems, Inc., 3125 Woodlark, Fort Worth, TX 76123; 817/294-3651
CIRCLE 340 ON READER SERVICE CARD

A software library of initialization and driver routines that allows applications programs written in BASIC, C, or assembly language to link easily to STD bus I/O cards is now available from **Pro-Log Corporation**. Designated **STD LIB 1.1**, the library supports Pro-Log's STD DOS system, which is MS-DOS 3.1 merged with the industrial quality hardware of Pro-Log's STD bus. STD LIB 1.1, based on the Microsoft Library Manager and Linker, consists of modules written in assembly language for high-speed execution and efficient use of memory. Release 1.1 supports the Pro-Log 7312-1/7314A-1 dual and quad RS-232 communications controller card, the Pro-Log 7507/7508 general purpose I/O card, the Pro-Log 7303 keyboard display card, and the Analog Devices RTI-1225 analog I/O card. The library specifications allow the user to add modules for I/O cards not currently supported. $295.
Pro-Log Corporation, 2560 Garden Road, Monterey, CA 93940; 408/372-4593
CIRCLE 333 ON READER SERVICE CARD

A language-independent, PC-based screen management system and software development environment has been introduced by **Master Computer Systems, Inc.** Aimed at professional developers of interactive transaction processing systems for DOS environments, **FORMIX version 2** automates 45 additional screen functions in a panel-oriented environment and provides full keyboard and system-control configuration. It features a memory-resident executive to manage all screen processing on behalf of the application program and a menu-driven environment on PC workstations to reduce development time. The objectives of FORMIX are to reduce program development cost, to optimize application screen performance, and to provide a flexible end-user interface. The FORMIX Executive (which also runs under TopView) executes 57 screen management functions. $695.
Master Computer Systems, Inc., 9531 W. 78th Street, Eden Prairie, MN 55344; 612/944-5220
CIRCLE 341 ON READER SERVICE CARD

The material that appears in Tech Releases is based on vendor-supplied information. These products have not been reviewed by the PC Tech Journal *editorial staff.*



Hercules compatible graphics: 720x348.

Plantronics compatible graphics: 640x200x4 colors.

Introducing The Paradise Hi-Res Graphics Card.™ Two Upgrades For The Price Of One.

Now you can have much higher resolution than your current video card gives you. Without buying new software. Or an expensive monitor.

All you need is the new Paradise Hi-Res Graphics Card.

It gives you the highest resolution possible on a standard monitor. With monochrome displays, you get Hercules compatible 720x348 graphics. With RGB and composite displays, you get Plantronics compatible graphics: 16 colors in 320x200, and 4 colors in 640x200. That's two upgrades for the price of one.

And no matter which monitor you choose, the Paradise Hi-Res Graphics Card gives you full compatibility with standard video modes.

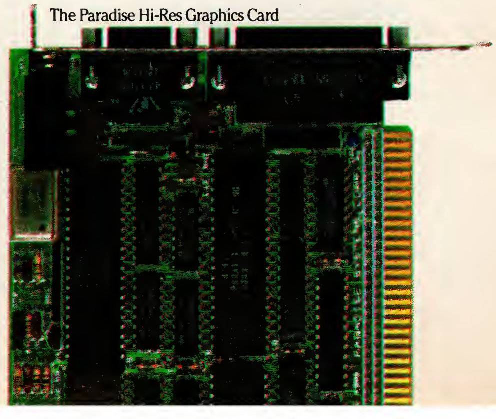
Better yet, you can use your existing software to its best advantage. Because Hercules and Plantronics graphics are widely supported by packages like Lotus 1-2-3, Symphony, Framework, and lots more.

And Paradise gives you all this, plus a parallel port. On one short card.

If you'd like to know more, we'll send you full information. Including a free Hi-Res software compatibility guide. Just call (800) 527-7977, Ext. 338, toll-free. From California, (800) 822-2020, Ext. 338. Or check it out at the best PC dealers.

PARADISE
SYSTEMS, INC

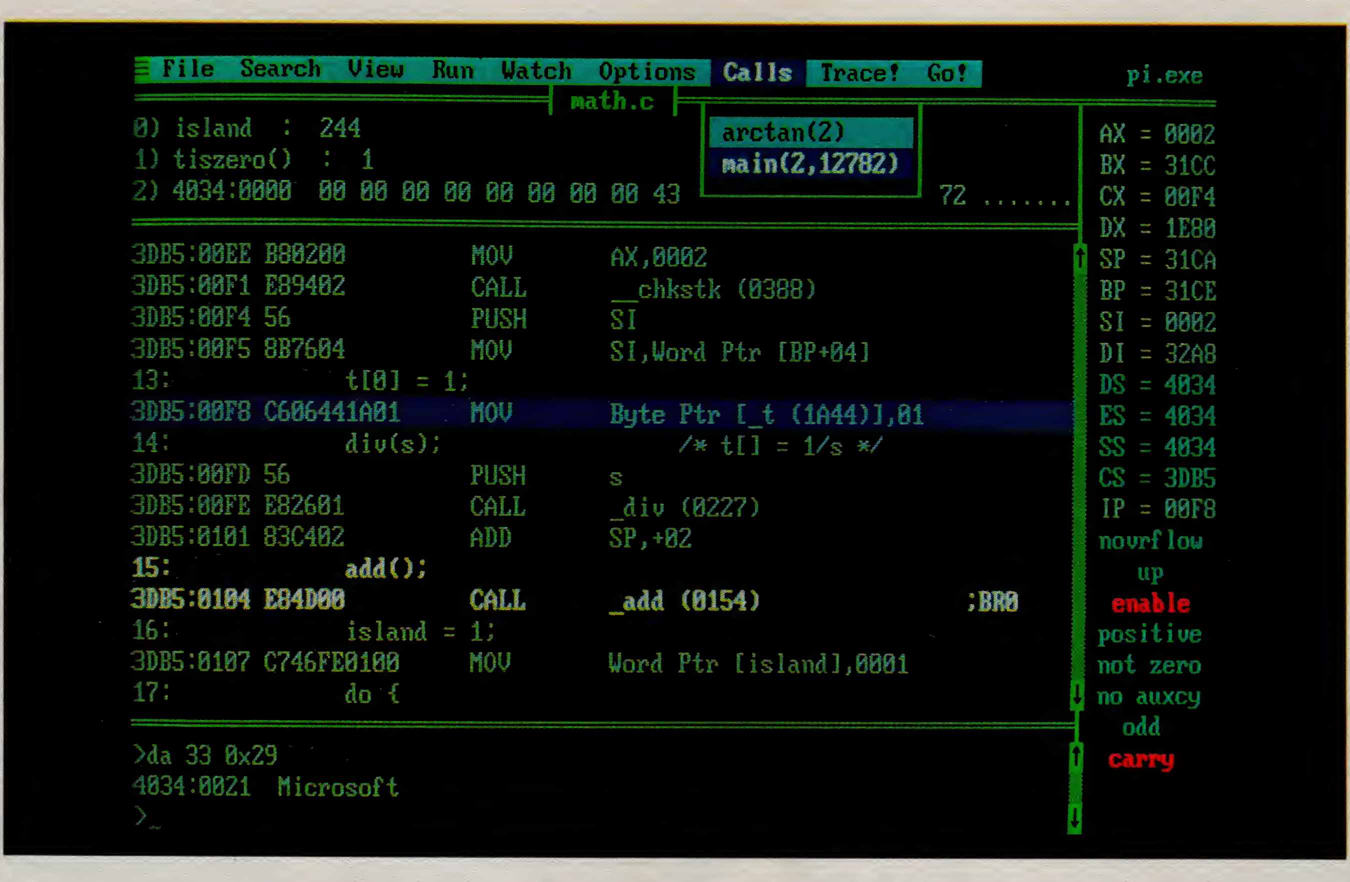
The Paradise Hi-Res Graphics Card



Trademarks: Paradise, Hi-Res Graphics Card-Paradise Systems Inc.; Lotus 1-2-3, Symphony-Lotus Development Corporation; Symphony, Framework-Ashton-Tate, Inc. Registered trademark: IBM-International Business Machines Corp.

CIRCLE NO. 230 ON READER SERVICE CARD

you've ever seen.



both at the same time. Open a window to view CPU registers and flags. Watch local and global variables as well. All while your program is running.

CodeView gives you complete control. Trace execution a line at a time—using source or assembly code. Or set conditional breakpoints on variables, memory or expressions. CodeView supports the familiar SYMDEB command syntax, as you'd expect. Commands are also available through drop-down menus. Combine the new window-oriented interface with our on-line help and debugging has never been easier. Or quicker.

Take the $5 CodeView tour.

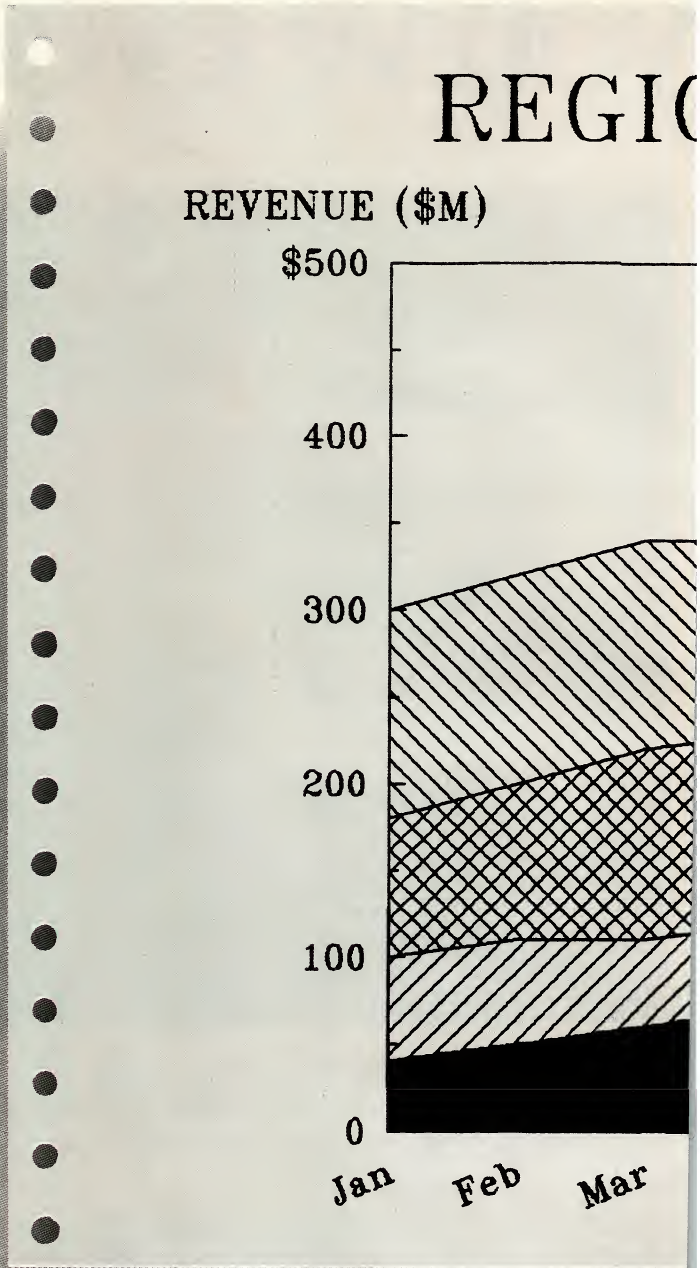
You may find it hard to believe our debugger can do all we've claimed. So we're offering test drives. Five bucks will put you behind the wheel of a Microsoft C demo disk with CodeView. See for yourself how fast debugging can get.

For more information about the CodeView demo disk, the new Microsoft C Compiler, a list of third party library support or the name of your nearest Microsoft dealer, call (800) 426-9400. In Washington State and Alaska, (206) 882-8088. In Canada call (416) 673-7638.

Microsoft® C Compiler

The High Performance Software

NEW DASH FOR YOU



Your old printer should look so good.

Introducing the ImagEnhancer™ PC add-on board. It enables ordinary monochrome or color dot matrix printers to produce high-quality, plotter-like graphics.

THE OUTPUT IS OUTSTANDING.

The ImagEnhancer is the perfect way to get presentation-quality graphs, charts and artwork from your old dot matrix printer.

What's more, it comes with a built-in 512K memory. So while your printer is printing one job, you can use your PC for another. Which will improve your productivity tremendously.

And of course, the ImagEnhancer works with the same computer products most everyone works with.

IBM® PCs, XTs, ATs or 100% compatibles with graphics capability. Alps, Epson® or Epson-compatible monochrome or color dot matrix printers. And Houston Instrument (DM/PL) plotter-compatible software, including Lotus® 1-2-3,® Symphony,® PFS®:Graph and Framework,™ among others.

The ImagEnhancer PC add-on board brings plotter-like output to dot matrix printers.

AND THE PRICE IS RIGHT.

At $595,* the ImagEnhancer costs a fraction of what you'd pay for a new plotter. It's simple to install, too. And backed by a one-year warranty.

The ImagEnhancer is a product of Alps America, a U.S. subsidiary of Alps Electric Co., Ltd. of Japan, an International Fortune 500 company. Other Alps America products include a full line of high-quality printers.

For more information, print samples, or the name of the dealer/distributor nearest you, write to Alps America, 3553 North First Street, San Jose, CA 95134. Or call **(800) 828-ALPS.** In California, (800) 257-7872. In Canada, (800) 858-2577.

Better yet, send in this coupon along with your payment. And teach your old dot matrix some new tricks.

CIRCLE NO. 247 ON READER SERVICE CARD

JAMES PARSLY

Using ROM Characters

Tic marks, plot titles, and y-axis titles can be labeled easily using the routines presented here.

Characters can be printed at any point in the 640-by-200 high-resolution graphics grid. A pattern table for the first 128 ASCII codes is found at location F000:FA6E in ROM. Each pattern consists of eight bytes representing the rows of dots that make up a character. In high-resolution graphics, the ROM BIOS displays characters by copying the appropriate bytes from the pattern table into video RAM.

The BASIC program called GLABEL.BAS (see listing 1) contains three subroutines that use this table for enhanced character displays. The first routine prints characters anywhere on the 640-by-200 high-resolution graphics grid and can be used for labeling tic marks. The second routine prints double-width characters similar to the ones that are possible with medium-resolution graphics. It can be used to label plot titles. The third routine prints characters rotated by 90 degrees and is useful for labeling *y*-axis titles. The main program offers a demonstration of the three routines.

James Parsly is a systems analyst for the Tennessee Valley Authority. He has a B.S. degree in computer science and mathematics.

LISTING 1: GLABEL.BAS

```
10 SCREEN 2:KEY OFF:CLS
20 REM
30 REM   DEMONSTRATE VERTICAL RESOLUTION
40 REM
50 X=0:FOR Y = 7 TO 20:L$=CHR$(Y+58):GOSUB 400
90 X=X+8
95 NEXT Y
100 REM
120 REM   DEMONSTRATE HORIZONTAL RESOLUTION
130 REM
140 Y = 60:FOR X = 65 TO 75:L$=CHR$(X):GOSUB 400
170 Y = Y + 8
175 NEXT X
180 REM
200 REM   DEMONSTRATE LARGE LETTERS
210 REM
220 Y=80:A$="BIG LETTERS"
230 FOR I = 1 TO LEN(A$)
250 X = 150+I*16:L$=MID$(A$,I,1)
260 GOSUB 690
280 NEXT I
290 REM
300 REM   DEMONSTRATE 90 DEGREE ROTATION
310 REM
320 A$="90 DEGREE ROTATION":X = 400
330 FOR I = 1 TO LEN(A$)
350 Y = 200-I*8:L$=MID$(A$,I,1)
360 GOSUB 870
380 NEXT I
390 LOCATE 20,1:STOP
400 REM
410 REM   SUBROUTINE PRINTS A LETTER L$ WHOSE LOWER LEFT-HAND CORNER
420 REM   WILL BE LOCATED AT (X,Y).
430 REM
440 IF ASC(L$) > 127 THEN RETURN
450 XX = X MOD 640:IF XX < 0 THEN XX = XX + 640   'MAKE 0 <= XX <= 639
460 REM
470 REM   UNLESS (X MOD 8) = 0, PARTS OF THE LETTER WILL BE IN TWO
480 REM   DIFFERENT COLUMNS.
490 REM
500 SHIFT = 2^(8-(XX MOD 8))                     'USED TO SHIFT BITS
510 LL = INT(XX/8)                               '1ST COLUMN
520 NN = (LL+1) MOD 80                           '2ND COLUMN
530 FOR II=Y-7 TO Y                              'ROW LOOP
540 YY = II MOD 200:IF YY < 0 THEN YY = YY + 200 'MAKE 0 <= YY <= 100
550 DEF SEG = &HF000                             'SEGMENT OF ROM
560 REM
570 REM   GET A BYTE REPRESENTING ONE OF THE EIGHT ROWS OF DOTS THAT
580 REM   MAKE UP THE CHARACTER.  SHIFT THE VALUE SO THAT THE BITS
590 REM   THAT GO IN THE TWO COLUMNS ARE SEPARATED INTO SEPARATE BYTES
600 REM
610 KK = PEEK(&HFA6E+ASC(L$)*8+II-Y+7)*SHIFT
620 DEF SEG = &HB800                               'SEGMENT FOR VIDEO
630 IF YY MOD 2 = 0 THEN DISP=0 ELSE DISP = &H2000 'OFF. FR EVN/ODD ROWS
640 MM = INT(YY/2)*80                              'OFFSET TO ROW
650 POKE DISP+MM+LL,PEEK(DISP+MM+LL) OR INT(KK/256) 'SET BITS IN 1ST COL
660 POKE DISP+MM+NN,PEEK(DISP+MM+NN) OR (KK-INT(KK/256)*256)  '& 2ND COL
670 NEXT II
680 RETURN
690 REM
700 REM   SUBROUTINE PRINTS A DOUBLE-WIDTH CHARACTER L$ WHOSE
710 REM   LOWER LEFT-HAND CORNER IS (X,Y)
720 REM
730 IF ASC(L$) > 127 THEN RETURN
740 DEF SEG =&HF000                               'SEGMENT OF ROM
750 FOR II = Y-7 TO Y                             'ROW LOOP
760 XX = X MOD 640:IF XX < 0 THEN XX = XX + 640   'MAKE 0<= XX <= 639
770 YY = II MOD 200:IF YY < 0 THEN YY=YY+200      'MAKE 0<= YY <= 199
780 KK = PEEK(&HFA6E+ASC(L$)*8+II-Y+7)     'GET PATTERN FOR A ROW
790 FOR LL=1 TO 8                          'LOOK AT BITS IN THE PATTERN
800 MM = KK MOD 2                          'GET LAST BIT
810 NN=XX+16-2*LL
820 IF MM <> 0 THEN PSET(NN MOD 640,YY):PSET((NN+1) MOD 640,YY) '2 DOTS
830 KK = INT(KK/2)                                'SHIFT BITS RIGHT
840 NEXT LL
850 NEXT II
860 RETURN
870 REM
880 REM   SUBROUTINE PRINTS A CHARACTER L$ WHICH HAS BEEN ROTATED
890 REM   90 DEGREES AROUND ITS LOWER LEFT-HAND CORNER (X,Y).
900 REM
910 IF ASC(L$) > 127 THEN RETURN
920 DEF SEG =&HF000                     'ROMS SEGMENT
930 FOR II = 1 TO 8                     'ROW LOOP
940 KK = PEEK(&HFA6E+ASC(L$)*8+II-1) 'GET PATTERN FOR A ROW
950 FOR LL=1 TO 8                       'COLUMN LOOP
960 MM = KK MOD 2                       'GET LAST BIT
970 NN=(X-16+2*II-1) MOD 640:IF NN < 0 THEN NN=NN+640  'GET X COOR.
980 YY = (Y-8+LL) MOD 200:IF YY < 0 THEN YY = YY + 200 'GET Y COOR.
990 IF MM <> 0 THEN PSET(NN,YY):PSET(NN+1,YY)           'MAKE TWO DOTS
1000 KK = INT(KK/2)                     'SHIFT BITS RIGHT
1010 NEXT LL
1020 NEXT II
1030 RETURN
```

To Over 30,000 Installations, MultiLink® MEANS Multi-User.

At over 30,000 sites, as many as one-quarter of a million users tap into the power of MultiLink® Advanced everyday.

Since 1981, they've come to rely on our multi-tasking, multi-user system for compatibility with their favorite software, and the ability to share disks, files, printers, and programs in a *true* PC-DOS environment.

From the largest of the Fortune 500 to the smallest in small business, MultiLink® has provided a cost-effective multi-user solution that's available from no one else.

MultiLink® Means Cost-Effective Timesharing on a PC. MultiLink® Advanced utilizes the principle of timesharing by sharing a central PC's peripherals, files, and processor time among nine users. Up to eight inexpensive terminals can be connected to a single non-dedicated IBM PC, XT, AT or 100% compatible using standard RS-232 ports. Each terminal effectively emulates a PC having up to 512K RAM.

PC-Shadow™ Workstations, shown below, even have an AT look-alike, as well as work-alike, keyboard, display, and serial port. In addition, password-protected remote access via modem can be made with either dumb terminals or PCs running our terminal emulation software.

MultiLink® Means PC-DOS Compatibility with a *Software-Driven* System. Lotus 1-2-3, Symphony, WordStar, dBASE III, & Multimate are just a sampling of the wide variety of PC-DOS software that's fully compatible.

Our software-driven system is also IBM NETBIOS compatible, so programs that are written for IBM's Token Ring will run on our multi-user system, as well.

MultiLink® Means Multi-User to Leading Computer Publications. Whether you read *PC Magazine*, "MultiLink® Advanced delivers on...convenience, speed, and flexibility," or *InfoWorld*, "If you want a low-cost multiuser system with up to eight terminals, MultiLink® Advanced is worth a serious look," it becomes clear that MultiLink® Advanced is a formidable contender in the multi-user marketplace.

See What MultiLink® Can Mean to You. Learn, firsthand, how our multi-user system can benefit your company. Call The Software Link TODAY for complete information and the authorized dealer nearest you. MultiLink® Advanced is $495 and comes with a money-back guarantee.

MultiLink®

ADVANCED

THE SOFTWARE LINK, INC.

8601 Dunwoody Place, Suite 632, Atlanta, GA 30338 Telex 4996147 SWLINK

CALL: 404/998-0700

Dealer Inquiries Invited

THE SOFTWARE LINK, INC./CANADA
250 Cochrane Drive, Suite 12 Markham, Ontario L3R 6B7
CALL: 416/477-5480

CIRCLE NO. 196 ON READER SERVICE CARD

PROGRAMMER DEVELOPMENT TOOLS FOR THE IBM-PC/XT/AT and compatibles.

See previous page for more information.

apl language

APL*PLUS/PC System by STSC		595	449
APL*PLUS/PC Tools Vol 1 by STSC		295	239
APL*PLUS/PC Tools Vol 2 by STSC		85	69
APL*PLUS Spreadsheet Manager by STSC	*New*	195	159
APL*PLUS/UNX System *For AT Xenix* by STSC		995	795
Btrieve *ISAM File Mgr with No Royalties* by SoftCraft		250	195
Financial/Statistical Library by STSC		275	219
Pocket APL by STSC		95	79
QNIAL *Combination of APL with LISP* by NIAL Systems		375	359
STATGRAPHICS *Statistical Graphics System* by STSC		795	619

arity products

Arity Expert System Development Package	*New*	295	279
Arity File Interchange Toolkit	*New*	50	48
Arity Prolog Compiler & Interpreter	*New*	795	739
Arity Prolog Interpreter	*New*	350	329
Arity SQL Development Package	*New*	295	279
Arity Screen Design Toolkit	*New*	50	48
Arity Standard Prolog	*New*	95	89

artificial intelligence

ExpertEDGE by Human Edge		795	659
Experteach *Complete System* by Intelliware		475	389
EXSYS *Expert System Development Software* by EXSYS		395	339
First Class by Human Edge	*New*	500	419
GCLISP *Golden Common LISP* by Gold Hill		495	CALL
GCLISP 286 Developer *LM Interpreter & LM Compiler*		1190	CALL
Insight 1 *AI Primer* by Level Five Research		95	75
Insight 2+ by Level Five Research	*New*	485	389
Logic-Line Series *All varieties* by Thunderstone	*New*	CALL	CALL
Microsoft LISP *Common LISP*		250	189
PLA microProlog by Programming Logic Associates		250	219
with APES		450	399
PLA Professional microProlog by Programming Logic		395	349
with APES		695	599
QNIAL *Combines APL with LISP* by NIAL Systems		375	359
Turbo Prolog Compiler by Borland International		100	79

assembly language

8088 Assembler *w/Z-80 Translator* by 2500 AD		100	89
ASMLIB *Function Library* by BC Associates	*New*	149	129
Cross Assemblers *Over 25 Varieties* from 2500 AD		CALL	CALL
Microsoft Macro Assembler *with utilities*		150	99
Turbo EDITASM *Fast Assembler* by Speedware		99	84
Visible Computer: 8088 by Software Masters		80	65

basic language

BetterBASIC by Summit Software		200	165
8087 Math Support		99	85
Btrieve Interface		99	85
C Interface	*New*	CALL	CALL
Run-time Module		250	225
Microsoft QuickBASIC Compiler	*New version*	99	79
Professional BASIC by Morgan Computing		99	79
8087 Math Support		50	47
True Basic		150	105
Run-time Module	*Special Price*	500	199

blaise products

Asynch Manager *Specify for C or Pascal*		175	136
C Tools Combination Package *Both Items Below*		175	149
C Tools		125	105
C Tools 2		100	84
Exec *Program Chainer*		95	79
Pascal Tools Combination Package *Both Items Below*		175	149
Pascal Tools		125	105
Pascal Tools 2		100	84
Turbo ASYNCH *for Turbo Pascal*		100	84
Turbo POWER TOOLS *for Turbo Pascal*		100	84
View Manager *Specify for C or Pascal*		275	209
with Source Code		295	239

borland products

REFLEX *Data Base System*		150	119
REFLEX Workshop	*New*	70	59
REFLEX and REFLEX Workshop *Combination Package*	*New*	200	159
Turbo DATABASE TOOLBOX		55	38
Turbo EDITOR TOOLBOX		70	54
Turbo GAMEWORKS TOOLBOX		70	54
Turbo GRAPHIX TOOLBOX		55	38
Turbo LIGHTNING		99	75
Turbo PASCAL		70	49
with 8087 or BCD		110	77
with 8087 and BCD		125	84
Turbo Prolog Compiler		100	79
Turbo TUTOR *for Turbo PASCAL*		35	28

c compilers

C-86 Optimizing Compiler		395	289
Datalight C Compiler *Small Memory Model*		60	49
Datalight Developer's Kit *with Large Memory Model*		99	79
DeSmet C Compiler *with Source Debugger*		159	145
Large Case Option	*New*	50	48
Eco-C *Complete Development System* by Ecosoft		125	89
Lattice C Compiler from Lattice	*See Lattice Section*	500	299
Let's C Compiler by Mark Williams		75	59
with csd Source Level Debugger		150	118
Microsoft C Compiler *with Source Debugger*	*New version*	CALL	CALL
MWC-86 by Mark Williams		495	299
Wizard C Compiler *Includes Lint* by Wizard Systems		450	369

c interpreters

C-terp by Gimpel Software *Specify compiler interface*		300	239
Instant C by Rational Systems	*New version*	500	379
Introducing C by Computer Innovations		125	105
Run/C by Age of Reason		150	99
Run/C Professional by Age of Reason		250	184

c utilities

Also refer to Blaise, Computer Innovations, Lattice, Microsoft, Phoenix, Polytron, SoftCraft and Xenix System V sections.

APT Application Programmer's Toolkit by Shaw American		395	339
Basic C Library by C Source		175	135
C Essentials by Essential Software		100	85
C to dBase *ISAM Manager* by Computer Innovations		150	139
c-tree *ISAM File Manager with source* by FairCom		395	329
C Utility Library by Essential Software		185	139
C Windows by Syscom		100	89
C Wings by Syscom		50	45
CI Probe *Source Level Debugger*	*Limited Quantity Sale*	225	159
CI RomPac *for C-86* by Computer Innovations		195	149
dBx *dBase to C Translator* by Desktop AI	*New*	350	329
dbVISTA *Single-User DBMS* by Raima		195	159
with Source Code		495	429
dbVISTA *Multi-User DBMS* by Raima		495	429
with Source Code		990	849
Entelekon Combo Package *All 3 items below*		200	175
C Function Library		130	115
C Windows		130	115
Superfonts for C		50	45
Essential Graphics *No Royalties* by Essential Software		250	219
Flash-up Windows by Software Bottling of NY		75	69
GraphiC *Mono version 2.2* by Scientific Endeavors		280	219
GraphiC *Color version 3.0* by Scientific Endeavors		350	299
The Greenleaf Functions by Greenleaf Software		185	135
Greenleaf Comm Library by Greenleaf Software		185	135
The HAMMER by OES Systems		195	175
MetaWINDOWS by Metagraphics		185	139
MetaWINDOWS/Plus by Metagraphics		235	199
Multi-Halo *with Royalties* by Media Cybernetics		300	219
On-line Help from Opt-Tech Data Processing		149	119
PANEL by Roundhill *Library Source Also Available*		295	229
PC Lint by Gimpel Software		139	109
Scientific Subroutine Library for C by Peerless		175	139
Vitamin C by Creative Programming		150	139
VC Screen *Interactive Forms Designer*		99	85
Zview *with Free Updates* by Data Mgmt Consultants		245	199

cobol language

Micro Focus COBOL Workbench		4000	3599
Micro Focus Level II COBOL	*Special Price*	1500	CALL
COMATH		200	169
FORMS-2		300	269
Level II Animator	*Special Price*	900	CALL
Level II SOURCEWRITER	*Special Price*	2000	CALL
Micro Focus Level II COBOL *For Novell NetWare*	*New*	2000	1799
Micro Focus Micro/SPF		175	159
Micro Focus Professional COBOL		3000	2395
Multi-user Runtime *for PC Network*	*New*	500	449
Microsoft COBOL	*See Microsoft Section*		
OPT-Tech Sort *Also Sorts Btrieve Files*		149	119
Realia COBOL		995	795
RM/COBOL by Ryan-McFarland		950	675
RM/COBOL 8X *ANSI 85 COBOL* by Ryan-McFarland		1250	995

debuggers & profilers

Advanced Trace-86 *with ASM Interpreter* by Morgan		175	139
CI Probe *Source Level Debugger*	*Limited Quantity Sale*	225	159
Codesifter *Execution Profiler* by David Smith	*New*	119	99
Codesmith-86 *Debugger* by Visual Age		145	109
Periscope I *w/Board & Switch* by Data Base Decisions		295	249
Periscope II *w/NMI Breakout Switch Only*		145	115
Periscope II-X *Software only*	*Special Price thru August*	115	85
The PROFILER *with Source Code* by DWB ssociates		125	95

forth language

CFORTH *Native Code Application Compiler* by LMI	*New*	300	239
LMI Forth/83 Metacompiler *Specify Target Processor*	*New*	750	599
PC/Forth by Laboratory Microsystems		150	119
PC/Forth+ by Laboratory Microsystems		250	209
Advanced Color Graphics Support	*New*	100	79
Enhanced Graphics Support	*New*	200	159
Intel 8087 Support	*New*	100	79
Interactive Symbolic Debugger	*New*	100	79
Native Code Optimizer	*New*	200	159
PCTERM *Modem Program for Smartmodem*	*New*	100	79
Software Floating Point	*New*	100	79

8·86

fortran language

ACS Time Series by Alpha Computer Service	495	429
Btrieve *ISAM File Manager* . . . See SoftCraft Section		
50 MORE: FORTRAN by Peerless Engineering	125	99
For-Winds by Alpha Computer Service	90	79
Forlib-Plus by Alpha Computer Service	70	55
Grafmatic or Plotmatic by Microcompatibles *New*	135	119
Grafmatic and Plotmatic by Microcompatibles *New*	240	219
Microsoft Fortran	350	209
Multi-Halo *with Royalties* by Media Cybernetics	300	219
PANEL *Screen Designer* by Roundhill	295	229
RM/Fortran by Ryan-McFarland	595	395
Scientific Subroutine Library by Peerless	175	139
The Statistician by Alpha Computer Service	295	259
Strings & Things by Alpha Computer Service	70	55

lattice products

Lattice C Compiler from Lattice	500	299
with Library Source Code	900	549
C Cross Reference Generator	50	39
with Source Code	200	159
C-Food Smorgasbord *Function Library*	150	99
with Source Code	300	195
C-Sprite *Source Level Debugger*	175	139
Curses *Screen Manager*	125	98
with Source Code	250	194
dBC *dBase File Manager for C*	250	194
with Source Code	500	388
LMK *Make Facility*	195	149
RPG II Compiler *No Royalties*	750	595
SecretDisk *File Encryption Utility*	60	49
SideTalk *Resident Communications*	120	95
Text Mgmt Utilities *(GREP,DIFF,ED,WC,Extract,Build)*	120	95
TopView Toolbasket *Function Library*	250	194
with Source Code	500	388
Z-80 C Cross Compiler	500	388
with Library Source Code	1000	789

microsoft products

Microsoft BASIC Interpreter for Xenix	350	279
Microsoft C Compiler *with source debugger* . . . New version	CALL	CALL
Microsoft COBOL Compiler	700	495
for Xenix	995	795
Microsoft COBOL Tools *with COBOL Source Debugger*	350	209
for Xenix	450	359
Microsoft Fortran Compiler	350	209
for Xenix	495	389
Microsoft LISP *Common LISP*	250	189
Microsoft Macro Assembler *with utilities*	150	99
Microsoft Mouse *Bus Version*	175	149
Microsoft Mouse *Serial Version*	195	159
Microsoft muMath *Includes muSIMP*	300	195
Microsoft Pascal Compiler	300	195
for Xenix	495	389
Microsoft QuickBASIC Compiler . . . New version	99	79
Microsoft Sort	195	149
Microsoft Windows	99	74
Microsoft Windows Developer's Kit *New*	500	CALL

modula-2 language

MODULA-2/86 Compiler by Logitech	89	65
with 8087	129	105
with 512K	189	149
MODULA-2 Editor by Logitech	59	49
MODULA-2 Runtime Debugger by Logitech	69	59
MODULA-2 Source Package by Logitech	99	79
MODULA-2 Utilities Package by Logitech	49	45

other languages

CCS MUMPS *Single-User version* by MGlobal *New*	60	55
CCS MUMPS *Multi-User version* by MGlobal *New*	450	379
Janus/ADA C Pack by R&R Software	95	89
Janus/ADA D Pack by R&R Software	900	699
Methods *Smalltalk* by Digitalk	79	69
Smalltalk/V by Digitalk *New*	99	89
SNOBOL4+ by Catspaw *New*	95	84

other products

Dan Bricklin's Demo Program by Software Garden	75	65
FASTBACK *Backup Utility* by 5th Generation Systems	179	159
Interactive EASYFLOW by Haventree Software	150	129
SET:SCIL by System Engineering Tools	349	299
Source Print by Aldebaran Laboratories	139	119
SRMS *Software Revision Mgmt System* by Quilt Computing	125	109

phoenix products

Pfantasy Pac *(Pfinish,Pfix+,Plink+,Pmaker,Pmate,Ptel)*	1295	949
Pfinish *Performance Analyzer*	395	249
Pfix-86 Plus *Symbolic Debugger*	395	249
PforCe *Comprehensive C Function Library*	395	289
Plink-86 *Overlay Linker*	395	249
Plink-86 Plus *Enhanced Overlay Linker*	495	389
Pmate *Macro Text Editor*	225	149
Pre-C *Lint Utility*	395	249
Ptel *Binary File Transfer Program*	195	139

polytron products

Polytron C Beautifier	49	45
Polytron C Library I	99	79
Polytron PowerCom *Communications*	179	139
PolyLibrarian *Library Manager*	99	79
PolyLibrarian II *Library Manager*	149	119
PolyMake *UNIX-like Make Facility*	99	79
PolyOverlay *Overlay Optimizer*	99	79
PolyWindows Products *All Varieties*	CALL	CALL
PolyXREF *Complete Cross Reference Utility*	219	179
PolyXREF *Support for one language only*	129	109
PVCS *Polytron Version Control System*	395	329
PVMFM *Polytron Virtual Memory File Manager*	199	149

softcraft products

Btrieve *ISAM File Manager with No Royalties*	250	195
Xtrieve *Query Utility for Btrieve*	195	169
Rtrieve *Report Generator for Xtrieve*	85	79
Btrieve/N *for Networks*	595	465
Xtrieve/N *Query Utility for Btrieve/N*	395	299
Rtrieve/N *Report Generator for Xtrieve/N*	175	159

text editors

Brief from Solution Systems	195	CALL
Epsilon *Multi-tasking Emacs-like editor* by Lugaru	195	165
FirsTime for Turbo by Spruce Technology	75	69
KEDIT *Xedit-like editor* by Mansfield Software Group	125	109
PC/VI by Custom Software Systems *New*	149	129
SPF/PC by Command Technology Corp	195	165
Vedit by CompuView	150	115
Vedit Plus by CompuView	225	180
XTC *Text Editor with source* by Wendin	99	84

turbo pascal utilities

Also refer to Blaise, Borland and SoftCraft sections.

ALICE *Turbo Pascal Interpreter* by Software Channels	95	69
FirsTime for Turbo by Spruce Technology	75	69
Flash-up Windows by Software Bottling of NY	75	69
Multi-Halo *with Royalties* by Media Cybernetics	CALL	CALL
On-line Help from Opt-Tech Data Processing	149	119
Screen Sculptor by Software Bottling of NY	125	95
Turbo EXTENDER by TurboPower Software	85	69
Turbo Professional by Sunny Hill Software	70	49
TurboPower Utilities by TurboPower Software	95	84
TurboWINDOW by MetaGraphics	80	69

wendin products

Operating System Toolbox *Build your own OS*	99	84
PCUNIX *Operating System*	99	84
PCVMS *Operating System Similar to VAX/VMS*	99	84
XTC *Text Editor with Pascal Source Code*	99	84

xenix system v

Complete Xenix System by SCO *All 3 items below*	1295	1099
Xenix Development System	595	529
Xenix Operating System *Specify XT or AT*	595	529
Xenix Text Processing Package	195	179

xenix languages and utilities

APL*PLUS/UNX System *For AT Xenix* by STSC	995	795
Btrieve *ISAM File Manager* by SoftCraft	595	465
c-tree *ISAM File Manager with Source* by FairCom	395	329
dBx *dBase to C Translator w/source* by Desktop AI *New*	550	499
dbVISTA *Single and Multi User versions* by Raima *New*	CALL	CALL
Informix by Relational Database Systems	995	839
Informix4GL by Relational Database Systems *New*	CALL	CALL
InformixSQL by Relational Database Systems *New*	995	839
Lyrix by SCO	595	489
Micro Focus Level II Compact COBOL *For AT* *New*	1000	899
Forms-2 *New*	400	359
Level II ANIMATOR *New*	600	539
Microsoft Languages . . . See Microsoft Section		
RM/COBOL by Ryan-McFarland *New*	1250	995
RM/FORTRAN by Ryan-McFarland *New*	750	599
SCO Professional *Complete Lotus clone* by SCO	795	695

VISA *Prices are subject to change without notice.* Ohio customers please add 5% state sales tax. MasterCard

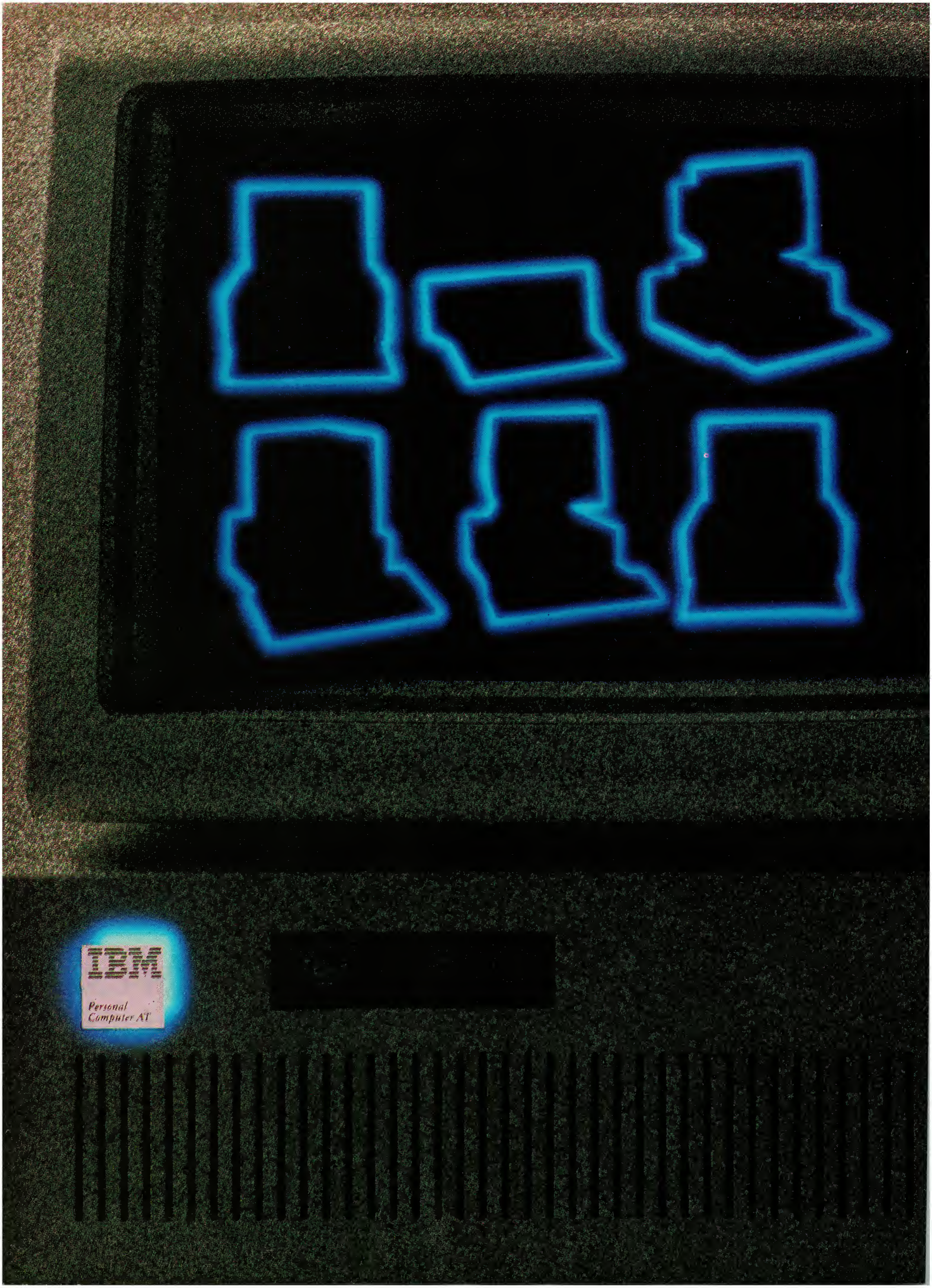
Hours: 8:30 AM to 8:00 PM EST, Monday through Friday.

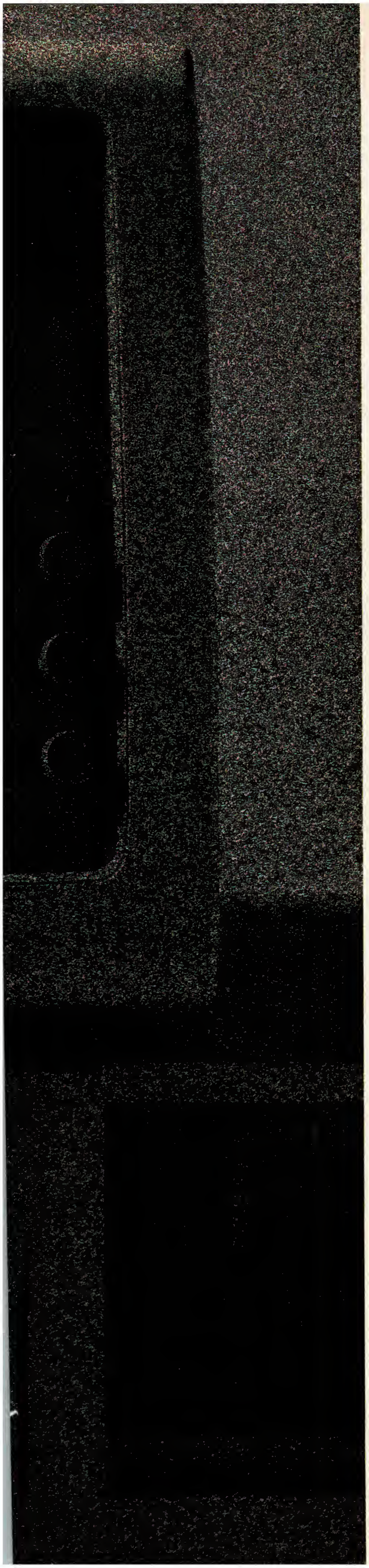
U.S.
1-800-336-1166
CANADA
1-800-225-1166
OHIO AND OVERSEAS
1-216-877-3781
CUSTOMER SERVICE **216-877-1110**

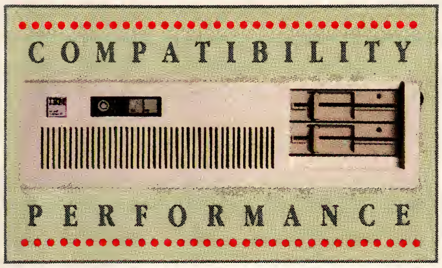
programmer's connection

136 SUNNYSIDE ST.
HARTVILLE, OHIO 44632

CIRCLE NO. 175 ON READER SERVICE CARD


IBM
Personal
Computer AT





Out from the Shadow of IBM...

...emerges an assortment of AT compatibles, each claiming to be better than the original. PC Tech Journal launches its series evaluating these AT alternatives one by one.

STEVEN ARMBRUST, TED FORGERON, and PAUL PIERCE

With the introduction of the PC/AT in 1984, IBM set the standard for high-performance personal computing. Since that time, a multitude of imitators has come out from the shadow of IBM, each claiming some combination of features, price, or performance as an advantage over the original. Because of the aggressive marketing practices of the manufacturers of these computers, and because several major players (such as AT&T, Hewlett-Packard, Tandy, and Sperry) are offering their own machines, IBM has been left with a smaller share of the market for 80286-based computers than it might have anticipated.

As sales figures indicate, AT compatibles are tempting alternatives to the original IBM version. However, finding the right computer can be an arduous task demanding much time and effort.

To assist in sorting through the choices, *PC Tech Journal* begins with this issue a series of articles on AT-compatible computers. Each article will focus on a single machine, examining in detail the two items most critical in evaluating an AT compatible: compatibility and performance. Each computer will be compared to the original 6-MHz IBM AT and to the 8-MHz AT released in April. A review of the Compaq Deskpro 286 on page 80 starts the series.

In keeping with the spirit of *PC Tech Journal's* continuing database management series in which products are evaluated according to the same exhaustive guidelines, and to ensure that each of the compatibles is evaluated in a consistent, detailed fashion, *PC Tech Journal* has devised a set of criteria for judging the machines. In addition, we have developed a set of com-

PHOTOGRAPH • MARC DAVID COHEN

patibility and performance tests that will be used to compare each computer with the IBM models.

Each article will conform to a template that will start out with a description of the computer's appearance and a listing of the type of equipment and options that are available with it. This section will contain a sidebar detailing the computer's vital statistics, including the base price, memory and disk-drive options, types of display adapters and monitors, number and kinds of expansion slots, and any extra features offered by the compatible that are not available with the IBM AT. These extra items—such as touch screens, tape drives, turbo switches, or any other feature that sets the machine apart—will be examined individually in the review.

The articles will look carefully at the standard features of the machine, such as the disk drives, display adapter, serial and parallel ports, and keyboard. It will report on unusual attributes of any item normally viewed as standard on an AT. Although all the computers will be tested with the manufacturers' standard hard disks, this section will mention whether any third-party disk controllers and hard disks have been certified for use with the computer.

Each review will venture inside the machine in question. The article will describe how easy (or how hard) it is to remove the cover of the system unit and install new boards or components. The power supply, disk drives, and memory will be discussed here.

Documentation provided with the computer will be reviewed and an overall evaluation made.

THE TESTS

At the heart of this series is a set of compatibility and performance tests. Two types of tests will be run. First, a selected set of commercial hardware and software products will be installed and run. The hardware includes an Intel Above Board AT, Microsoft bus and serial mouse, a Hayes Smartmodem 1200B, and an IBM Enhanced Graphics Adapter and monitor. The commercial software products chosen are Microsoft Windows 1.01 and Word 2.0 (to test a graphics-intensive environment and serial and parallel printer ports); Borland's SideKick 1.56A, SuperKey 1.11A, and Turbo Lightning 1.01A (to test memory-resident programs); Hayes's Smartcom II 2.1 (to test the communications interface); Living Video Text's Ready! 1.00D (to test the use of expanded memory as defined by the Lotus-Intel-Microsoft specification); and the IBM AT Advanced Diagnostics 1.01 (for a general checkup).

The second type of test involves running programs designed for this series. To test compatibility, these programs provide information about the processor, BIOS, hard-disk drive, and keyboard. To measure performance, they determine the speed of the processor, hard-disk drive, and memory.

Table 1 lists the results of these tests when run on three different IBM computers: an original AT with a 20MB hard disk, the second version of the AT with a 30MB hard disk (and BIOS that limits the clock speed to 6 MHz), and the latest version of the AT with an official 8-MHz clock and a 30MB hard disk.

The two items considered the most critical in evaluating an AT compatible are compatibility and performance. Each computer will be compared to the 6- and 8-MHz IBM AT models.

In each of the upcoming articles, the same tests will be run on the AT-compatible computer and the results compared with those of the 8-MHz AT.

The source code for the tests is listed here in abbreviated form; the complete listings are on PCTECHline. *[The AT Evaluation Suite is copyright © 1986 by Ted Forgeron, Steven Armbrust, Paul Pierce, and* PC Tech Journal. *All rights reserved; the resale, other commercial use of the programs for profit, or publication of the programs is prohibited. Any internal or noncommercial use or display of the programs or their results requires permission from* PC Tech Journal *and full credit to* PC Tech Journal *and the authors. —WF]*

The tests are divided into five separate programs: ATBIOS, ATKEY, ATPERF, ATFLOAT, and ATDISK.

ATBIOS. Written in Turbo Pascal, the ATBIOS code is shown in listing 1. This test examines ROM locations of the BIOS to determine the date, machine ID, and copyright statement. This information determines which version of BIOS is present in the computer and who manufactured the BIOS, an important consideration for compatibility.

ATBIOS looks at the BIOS data area (a 256-byte area of RAM starting at address 400H that the BIOS uses for its variable data) to retrieve information such as the number of diskette drives installed, whether the 80287 math coprocessor is present, the initial and current video modes, the DMA flag, the number of serial and parallel ports, and the keyboard buffer (the last few characters typed at the keyboard). The test looks for this information at the locations used by the AT's BIOS. If the compatible computer's BIOS uses different locations to record the same items, the test will return useless information for those particular items.

Although the ATBIOS program was written specifically for AT-compatible computers, it will run on both PC- and AT-style machines and will display the machine type ("AT COMPATIBLE" or "NOT AT") based on the value of the BIOS machine ID field.

ATKEY. ATKEY is an assembly language program that tests the programmability of the keyboard. The AT contains an 8042 processor that defines the characteristics of the keyboard. This chip can be reprogrammed to redefine the keys or change the way the keyboard operates. By programming the 8042, ATKEY tests whether utility programs, such as those that speed up the typematic action, work on the compatible.

The source code of ATKEY is shown in listing 2. ATKEY sends a command to the 8042 keyboard processor and then attempts to set the keyboard's repeat count to 30 characters per second. If the computer does not have an 8042 controlling the keyboard, the test does not receive an acknowledgement from the original command. In such a case, ATKEY returns an error message.

The Microsoft assembler is used to produce ATKEY's object code. Assuming that ATKEY.ASM contains the assembly language source code, the following commands are used to assemble and link the program:

```
masm atkey,;
link atkey,;
```

Because ATKEY tests functions that are specific to the AT, this test will not run on PC- or XT-compatible computers. Once the ATKEY program has been run, the keyboard repeat count is left at 30 characters per second. The system should be rebooted if the user finds this setting undesirable.

ATPERF. The ATPERF tests determine the clock rates of both the CPU and the math coprocessor, measures the speed of memory accesses, and determines

TABLE 1: *AT Compatibility and Performance Tests*

	AT with 30MB disk (8-MHz)	AT with 30MB disk (6-MHz)	AT with 20MB disk (original)
ATBIOS			
ROM BIOS date	11/15/85	06/10/85	01/10/84
ATPERF			
Average RAM instruction fetch (μs)	0.403 (100)[a]	0.569 (71)	0.555 (73)
Average RAM read time (μs)			
BYTE	0.401 (100)	0.565 (71)	0.549 (73)
WORD	0.401 (100)	0.565 (71)	0.549 (73)
Average RAM write time (μs)			
BYTE	0.401 (100)	0.566 (71)	0.549 (73)
WORD	0.401 (100)	0.565 (71)	0.549 (73)
Average ROM read time (μs)			
BYTE	0.401 (100)	0.565 (71)	0.549 (73)
WORD	0.401 (100)	0.565 (71)	0.549 (73)
Average video write time (μs) (CGA only)			
BYTE	1.208 (100)	2.143 (56)	2.141 (56)
WORD	2.415 (100)	3.822 (63)	3.831 (63)
Average EMM read time (μs)			
BYTE	0.402 (100)	0.566 (71)	0.549 (73)
WORD	0.402 (100)	0.566 (71)	0.549 (73)
Average EMM write time (μs)			
BYTE	0.402 (100)	0.566 (71)	0.549 (73)
WORD	0.402 (100)	0.566 (71)	0.549 (73)
CPU clock rate (MHz)	8 (100)	6 (75)	6 (75)
Math coprocessor clock rate (MHz)	5.3 (100)	4 (75)	4 (75)
Refresh overhead (%)	7.1	13	9.6
RAM read wait states	1	1	1
RAM write wait states	1	1	1
ROM read wait states	1	1	1
Video write wait states (CGA)	8	11	11
EMM read wait states	1	1	1
EMM write wait states	1	1	1
ATFLOAT			
Performance as percentage of 8-MHz AT	7	7	100
ATDISK			
Sectors/track	17	17	17
Heads	5	5	4
Cylinders	731	731	613
Total space (million bytes)	31.81	31.81	21.34
Track-to-track seek time (ms)	10.0	6.0	9.1
Average seek time (ms)	37.0	37.1	37.5
Effective transfer rate (KB/sec)	170.1	170.1	168.6
DOS file I/O (sec)	7.3	7.5	7.4
Interleave	3	3	3

[a]*Figures shown in parentheses represent the relative performance expressed as a percentage compared to* PC Tech Journal*'s baseline machine, the 8-MHz, 30MB AT.*

These results were taken from several machines and averaged. Variations are apparent in the results from the ATDISK test. Track-to-track seek time on the original IBM AT varied substantially (7.2 ms to 13.3 ms) from machine to machine. The average seek time also varied but by a much lower amount. The effective transfer rate varied even less. The variation can be accounted for by considering the manufacturing tolerances. The drives come from a variety of manufacturers and each will behave in a different way with temperature changes, for example. Because the effective transfer rate is relatively constant, this apparent variation is negligible. Note, however, that the numbers obtained from the new model AT were consistent from machine to machine, showing tighter manufacturing tolerances.

the number of wait states that occur for the various types of memory accesses. The program is written in Microsoft C with assembly language subroutines. In addition to listing the actual times for the various operations, ATPERF displays relative performance numbers that compare the machine being tested to the new 8-MHz AT (a value of 1 is equivalent to the new AT; a value of 2 is twice as fast; etc.) Listing 3 shows the C source code of this program, and listing 4 shows the assembly language subroutine TIME.ASM.

For all of its timings, ATPERF uses timer number 2 of the programmable interval timer (an Intel 8254-2 chip). This is the timer that is normally used with the speaker. The program assumes that the timer frequency is 1.193180 MHz, as documented in the data sheet for the Intel chip. To ensure this value is accurate, each tested computer will be programmed to sound a 1,000-Hz tone, which will be measured by connecting a microphone to a frequency meter calibrated to the National Bureau of Standards radio station WWV. If the measured frequency of timer number 2 is incorrect in the computer being tested, the value used in the program will be adjusted accordingly.

ATPERF performs all measurements in the same general fashion. The main program calls an assembly language subroutine that turns on timer number 2, executes several operations in sequence (usually 1,000), and turns off the timer. To minimize the loop overhead, the timed operations are coded 100 or more to a loop rather than having a single-instruction loop. The number of clock ticks is then returned to the main program where it is turned into actual time. The subroutine is called many times (usually 100) and the combined results are averaged.

ATPERF's first measurement calculates the processor clock rate by calling a subroutine named _MULTIME to take 100 measurements of 1,000 multiply instructions. Multiply instructions are used because each MUL instruction takes 21 clock cycles, much longer than the time needed to fill up the processor's instruction queue. Therefore, the measurements are based purely on the processor speed, not on the speed of fetching instructions from RAM.

The formula for calculating the amount of time for each processor clock cycle determines the amount of time for the operation (the number of timer ticks divided by the timer rate) and divides it by the number of processor clock cycles that occurred during

LANGUAGES THAT ARE CAUSING THE BIGGEST PROGRAMMING BACKLOG IN HISTORY ARE ALSO EATING NICE BIG HOLES IN OUR POCKETS.

Whether it's BASIC, COBOL, Pascal, "C", or a data base manager, you're being held back.

Held back because the language has frustrating limitations, and the programming environment isn't intuitive enough to keep track of what you're working on.

In the real world, there's pressure to do more impressive work, in less time, and for more clients.

We've been given some incredibly powerful hardware in recent times, but the languages aren't a whole lot better than they were 20 years ago.

So, whatever language you have chosen, by now you feel it's out to get you — because it is.

Sure, no language is perfect, but you have to wonder, "Am I getting all I deserve?"

And, like money, you'll never have enough.

Pretty dismal, huh?

We thought so, too.

So we did something about it.

We call it CLARION™.

You'll call it *"incredible."*

Distributed on 7 diskettes, CLARION consists of over 200,000 lines of code, taking 3+ years to hone to "world-class" performance.

With CLARION you can write, compile, run and debug complex applications in a New York afternoon.

Even if you're in Savannah.

It gives you the power and speed to create screens, windows and reports of such richness and clarity you would never attempt them with any other language.

Because *you* would have to write the code.

With CLARION you simply design the screens using our SCREENER utility and then CLARION writes the source code AND compiles it for you in seconds.

Likewise, you can use REPORTER to create reports.

Remember, only CLARION can recompile and display a screen or report layout for modification.

And with no time wasted.

All the power and facilities you need to write great programs, faster than you ever dreamed of.

Programs that are easy to use.

Programs that are a pleasure to write.

And to you that means true satisfaction.

You've coveted those nifty pop-up help windows some major applications feature. But you can't afford the time and energy it takes to write them into your programs.

That's the way it used to be.

So we fixed that, too.

CLARION's HELPER is an interactive utility that let's you design the most effective pop-up help screens that you can imagine. And they're "context sensitive," meaning you can have help for every field in your application.

Unlike the other micro languages, CLARION provides declarations, procedures, and functions to process dates, strings, screens, reports, indexed files, DOS files and memory tables.

SAY IT IN

CLARION™

A2TJ/3

1-800-354-5444

VISA AMERICAN EXPRESS MasterCard

Imagine making source program changes with the CLARION EDITOR. A single keystroke terminates the EDITOR, loads the COMPILER, compiles the program, loads the PROCESSOR and executes the program. It's that easy!

Our data management capabilities are phenomenal. CLARION files permit any number of composite keys which are updated dynamically.

A file may have as many keys as it needs. Each key may be composed of any fields in any order. And key files are updated whenever the value of the key changes.

Like SCREENER and REPORTER, CLARION's FILER utility also has a piece of the CLARION COMPILER. To create a new file, you name the Source Module. Then you name the Statement Label of a file structure within it.

FILER will also automatically rebuild existing files to match a changed file structure. It creates a new record for every existing record, copying the existing fields and initializing new ones.

Sounds pretty complicated, huh? Not with CLARION's documentation and on-line help screens. If you are currently competent in BASIC, Pascal or "C" you can be writing CLARION applications in a day. In two days you won't believe the eloquence of your CLARION programs.

Okay, now for the best part of all. You can say it in CLARION for $295.00—plus shipping and handling. All you need is an IBM® PC, XT, AT or true compatible, with 320 KB of memory, a hard disk drive, and a parallel port.

And we'll allow a full 30 day evaluation period. If you're not satisfied with CLARION, simply return it in its original condition for a full refund.

If you're not quite ready to take advantage of this no-risk opportunity, ask for our detailed 16 page color brochure. It vividly illustrates the elegance of CLARION. Consider it a preview of programming in the fast lane.

Either way, the 800 call's a freebie.

BARRINGTON SYSTEMS, INC. 150 EAST SAMPLE ROAD POMPANO BEACH, FLORIDA 33064 305/785-4555

CIRCLE NO. 105 ON READER SERVICE CARD

that time. An overhead factor is included in the number of processor clock cycles to allow for any extra time caused by the loop and turning on and off timer number 2.

The instruction fetch time is calculated using a similar formula. Instead of timing MUL operations, the instruction fetch test _WMOVTIME times MOV operations. These take only two clock cycles, which is also the minimum amount of time for an instruction fetch. The performance of this test depends not on how fast the processor can execute the instruction, but on the speed of the instruction fetch.

ATPERF calculates memory read and write times by using the processor's string-operation instructions, which move and store words or bytes. With the move instructions (MOVSB and MOVSW), bytes or words are read from DS:SI and written to ES:DI. The CX register specifies the number of bytes or words to move. With the store instructions (STOSB and STOSW), bytes or words are written from the AX register to ES:DI. As with the move instructions, the CX register indicates the number of times to store the value in AX, with DI being incremented each time.

The memory-write time is calculated directly from the store operations because they are pure write operations. The memory-read time is calculated using both the store time (which measures a write operation) and the move time (which measures both a read and write time). In fact, the store time is subtracted from the move time in order to obtain the read time.

ATPERF checks four kinds of memory: RAM, ROM, video memory (on a color/graphics adapter only), and EMM (expanded memory manager on the Intel Above Board). If no expanded memory is present in the computer, ATPERF skips the EMM checks. The program could be modified to check the video memory on the monochrome adapter by changing the hard-coded 0B800H value in the _WVIDTIME and _BVIDTIME routines to 0B000H.

For RAM and EMM memory, both read and write times are measured. Only the read time is measured for ROM (by using MOVSB and MOVSW to read from ROM and write to RAM, and then subtracting the time required for STOSB and STOSW to write to RAM). For video RAM, only the write time is measured, because rarely is any other kind of operation performed during normal system operation.

When checking memory speed, ATPERF measures the times for both byte and word accesses. This is done to check whether the computer truly contains 16-bit data paths. Where 16-bit data paths exist, the times for byte and word accesses are the same. For 8-bit data paths, the word accesses take twice as long as byte accesses.

Several assembly language routines (shown in listing 4) measure the memory access times. _BMVSTIME times MOVSB instructions using normal RAM; _WMVSTIME measures similar MOVSW instructions. _BROMTIME times MOVSB

Unlike ATPERF, which takes great pains to isolate the operations it measures, ATFLOAT measures the type of floating-point operation that would naturally occur in a real-world application.

instructions that transfer data from ROM to RAM; _WROMTIME measures similar MOVSW instructions. _BSTOTIME times STOSB instructions using normal RAM; _WSTOTIME is used to time similar STOSW instructions. _BVIDTIME and _WVIDTIME clock the video memory write times. _BEMSTIME, _WEMSTIME, _BEMMTIME, and _WEMMTIME measure the access times for expanded memory on the Intel Above Board.

Next, ATPERF calculates the math coprocessor (80287) clock rate. As with the EMM test, the program checks for the existence of a math coprocessor before proceeding. If no 80287 is present, this portion of the test is skipped.

Part of the math coprocessor calculation is analogous to the CPU clock rate test, in which the program calls _FPTIME to measure the time required to perform floating-point divide (FDIV) instructions. Because floating-point instructions take much longer to execute than other instructions, _FPTIME executes fewer loops than the other test routines are able to execute. If it did not, the 16-bit count in the timer would overflow during the test.

The final part of the math coprocessor calculation is similar to the CPU clock rate test; the amount of time taken to execute the instructions is divided by the number of math coprocessor clock cycles that occurred. Because the _FPTIME routine contains no looping, no loop overhead needs to be included in the calculation.

The ATPERF program also measures the refresh overhead. This value is the percentage of memory access time the computer uses in refreshing the memory (that is, rewriting it to prevent it from losing data).

Memory refresh is required for all computer systems that contain dynamic RAM. The cells in a dynamic RAM chip are essentially capacitors that form gates of a transistor. These capacitors are either charged or discharged to denote the 1s or 0s of the bits. If the memory is not refreshed, this charge leaks out of the capacitors in just a few milliseconds. In PCs, ATs, and any computer that uses the PC bus, a refresh signal occurs every 15 microseconds. This causes the computer to generate special refresh cycles, in which it rewrites all of the data contained in RAM.

In order to determine the refresh overhead, ATPERF uses the value it calculated for RAM read time and divides it by the value for a CPU clock cycle. This gives the number of clock cycles for the read operation. Because a RAM read operation should take an integral number of clock cycles, the fractional part of the result is due to memory refresh operations that occur during the 10,000 read operations of the test. This fraction of a memory cycle is then divided by the number of CPU cycles required for a read operation and is displayed as a percentage.

ATPERF's final test calculates the number of wait states that are associated with accessing the various types of memory. A wait state is an extra clock cycle during which the CPU remains idle while waiting for a response from memory. The CPU requires two clock cycles from the time it issues a memory read or write until it is ready to issue another one. If at that time, the memory has not sent a ready signal to indicate that the last read or write is finished, the processor idles for a clock cycle before it checks for the ready signal again. Each one of these extra clock cycles is a wait state.

The ANALYZE subroutine determines the number of wait states for the various types of memory by using the memory read and write times calculated earlier. It subtracts the amount of time required for two CPU clock cycles from the time required for the memory access. The result is divided by the time for a single CPU cycle to give the number of extra cycles (the wait states). A smaller number of wait states indi-

cates a higher level of performance, because the processor is wasting fewer clock cycles idling.

In addition, ANALYZE determines the width of the data path for the different memory types. It does so by comparing the times for the corresponding word and byte accesses. If these values are nearly the same, that memory is accessed an entire word at a time. If a word access takes approximately twice as long as a byte access, the data path is only one byte long.

When displaying the final results, ATPERF includes relative numbers that reflect how well the tested computer compares with the new 8-MHz AT in read and write times and clock rates. The program contains hard-coded values for the AT that were calculated using ATPERF. These numbers are divided by the calculated values to obtain a relative value.

The Microsoft C compiler and the Microsoft assembler are used to produce the object code of the ATPERF program. Assuming that ATPERF.C contains the C source code and TIME.ASM contains the assembly language source code, the following commands are used to compile and link the program:

```
msc atperf.c,;
masm time.asm/r,,;
link
   atperf + time,,,slibfp.lib + slibc.lib + em.lib;
```

The **/r** option on the **masm** command directs the assembler to generate floating-point instructions that can be executed on the 80287 coprocessor.

ATPERF is written specifically for AT-compatible computers. Running this program on PC-compatible computers will produce inaccurate results.

ATFLOAT. ATFLOAT, which is written in Microsoft C (listing 5), measures the performance of floating-point operations. Unlike ATPERF, which takes great pains to isolate the operations it measures, ATFLOAT measures the type of floating-point operation that would naturally occur in a real-world application. Instead of stringing together floating-point multiply instructions, ATFLOAT measures the time taken to perform 100 multiplies on a 20-by-20 matrix. Although the vast majority of the time is taken up by floating-point processing, the measurement also includes CPU time for looping through the matrices; this is time that would also be spent in any real floating-point application.

The test begins by filling the 20-by-20 matrix with random numbers. Then ATFLOAT performs 100 matrix multiplies while using the Microsoft C **time** function to record the number of seconds required. The result of the test is displayed as a relative number that compares the tested computer with a new 8-MHz AT that contains an 80287.

The Microsoft C compiler produces the object code of ATFLOAT. Assuming that ATFLOAT.C contains the C source code, the following commands are used to compile and link the program:

```
msc atfloat.c/FPi,,atfloat.cod;
link atfloat/map,,,slibfp.lib + slibc.lib;
```

The **/FPi** option on the **msc** command generates code that uses the 80287 if one is present in the computer. If the 80287 is not present, the Microsoft floating-point emulator is used. Running ATFLOAT on a computer without an 80287 takes as long as 15 minutes (as compared to a minute or so with the 80287).

> *The ATDISK program's measurement of the effective disk-transfer rate allows the interleave value to be determined empirically.*

ATFLOAT.COD, which is specified in the **msc** command, is a file in which the compiler places the assembly language instructions for the code that it produces. This file can be examined to see the floating-point instructions that are generated.

ATFLOAT will produce accurate results when run on both AT-compatible and PC-compatible computers.

ATDISK. ATDISK returns the characteristics of the computer's hard disks and measures two types of disk performance: one uses BIOS calls to isolate and time low-level disk operations; the other measures disk performance during normal system operations by using DOS function requests to perform standard file I/O. ATDISK is written in Microsoft C (listing 6), with assembly language subroutines listed in the file called PDISKIO.ASM (listing 7).

To measure the time that is required to perform the operations in this program, ATDISK uses the clock count maintained by the BIOS in a four-byte area of RAM beginning at absolute location 46CH. With each clock tick (approximately 18.2 times per second), the BIOS increments the value in this four-byte RAM area. The assembly language subroutine **a_tick** reads this data area before and after a test. The difference between the two values is the number of clock ticks that have occurred during a test. ATDISK then converts this number to elapsed time.

ATDISK's first type of performance check, determining the low-level performance characteristics of the disk (including how many disk drives are connected to the computer), uses the disk services of BIOS interrupt 13H. The subroutine **disk_io** is used to invoke these services three times, with the function request number being the first parameter of the subroutine:

- BIOS disk function request 8 (get current drive parameters) determines the size of the disk and the number of sectors, heads, and cylinders.
- Function request 12 (the BIOS seek function) measures the track-to-track seek time. The program seeks from sector 1, head 0, cylinder 0 to sector 1, head 0, cylinder 1 and back again. Therefore, it seeks between two adjacent tracks 500 times, for a total of 1,000 seek operations.
- The third measurement calculates the average seek time. This test measures the time taken to seek to 1,000 random locations on the disk. It divides the total seek time by the number of seeks to obtain the average seek time.

ATDISK's second performance check calculates the effective disk-transfer rate, which is the speed at which data can be transferred to or from a formatted disk. (This is different from the absolute disk-transfer rate, which is how fast a stream of bits can be transferred from the read/write head to the media or vice versa. The effective transfer rate includes the effects of disk formatting—that is, the interruption of data flow due to sectoring and interleaving. This is the rate that would occur during normal system operation.)

ATDISK measures the effective transfer rate by using the BIOS read sectors service (function 2 of interrupt 13) to read track 0 a total of 1,000 times. The rate is calculated by dividing the number of kilobytes read by the time taken to read that information.

Measuring the effective transfer rate of the disk allows the interleave value of the hard disk to be determined. The interleave is a measure of the spacing of logical, as opposed to physical, sectors on the disk.

Each disk is divided into a number of equal-sized sectors (usually 17 on an XT or AT hard disk), which are numbered sequentially from 1 to 17. These

are the *physical* sector numbers. The computer does not necessarily access the sectors in sequence, however. For example, if the computer writes a block of data that is longer than a sector, it might put the first sector of data in physical sector 1, skip sector 2, and put the next part of the data block in physical sector 3. The order in which the physical sectors are accessed to read or write sequential data is the *logical* numbering of the sectors.

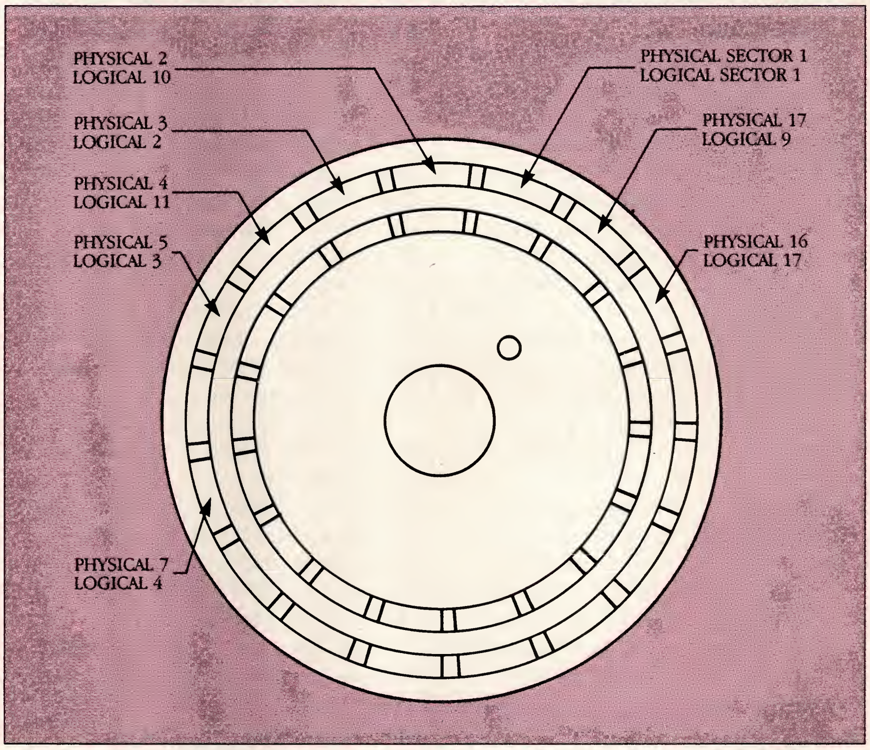
The interleave value, then, is the distance, in sectors, between two logical sectors. An interleave of 1 means that logical sectors are next to each other (in other words, logical and physical sector numbering is the same). An interleave of 2 means that there is a sector between two sequential logical sectors. An interleave of 3 means that two sectors occur between logical sectors. Figure 1 shows the logical sector numbering of a disk set up with an interleave value of 2; figure 2 shows an interleave value of 3.

The interleave value is important for overall disk performance. The disk is always spinning, so if the computer needs to access a sector that is not directly under the disk's read/write head, it must wait until that part of the disk moves around under the head again. With a judicious choice of interleave, the time required by the computer between sector accesses (either to dispose of the information it just read or to prepare the next information to write) is the same as the time required to spin the next logical sector under the read/write head.

An interleave value that is too large can hurt performance because the computer will be forced to wait longer for the appropriate sector to move under the head. Likewise, when the interleave value is too small, the sector will have passed under the read/write head before the computer is ready, causing the computer to wait almost a complete revolution until the sector moves under the head again.

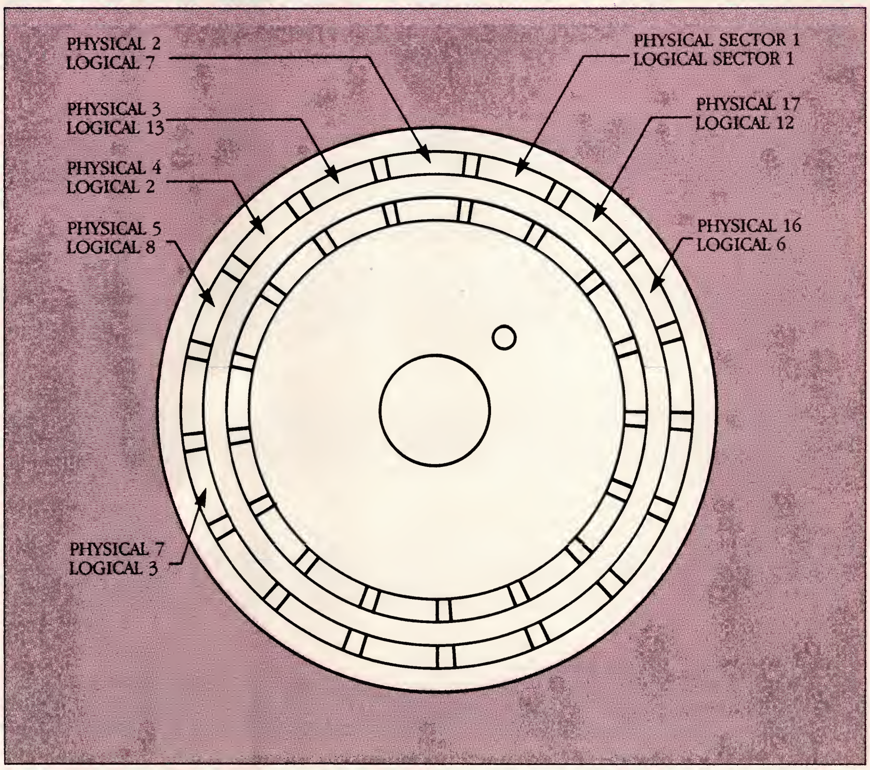
Although practically any interleave value can be selected for any disk drive, the optimum choice really depends on the speed of the processor and the performance of the drive. A higher-performance computer will tend to have a smaller optimum interleave than a lower-performance computer, and the smaller interleave will mean increased throughput. However, the way in which a disk controller manages the drive (and the data coming from or going to the drive) also can have a major effect on I/O performance. Therefore, a com-

FIGURE 1: *Interleave Value of 2*



When a disk has an interleave of 2, a logical sector is two physical sectors from the next logical sector. Logical sectors are numbered in the order they are accessed. Generally, a higher-performance computer has a smaller optimum interleave.

FIGURE 2: *Interleave Value of 3*



A disk with an interleave of 3 is not necessarily slower than one with an interleave of 2 because the optimum choice depends on the speed of the processor and the performance of the drive. If the value is too high or too low, the disk has to wait a long time before the next sector is moved under the head.

puter with an optimum interleave of 2 might not always exhibit better disk performance than a computer that has an optimum interleave of 3.

The hard disks on both the 6- and 8-MHz ATs use a default interleave value of 3. ATDISK's measurement of the effective disk-transfer rate enables the interleave of each AT-compatible computer to be determined empirically. This is accomplished by first running the ATDISK program when the hard disk is formatted for the default interleave, then rerunning the program after the hard disk has been reformatted for specific interleave values (such as 1, 2, and 3). Because the interleave affects the performance of I/O operations, the values for the DOS disk test will be different each time the interleave is changed. The default value can be determined by comparing the first result with the results of all the later tests.

ATDISK measures the performance of disk operations that occur during normal day-to-day activities by accessing the hard disk via DOS function requests. This test has two phases. During the first phase, ATDISK creates a file, writes 20KB of information to the file in 2KB increments, and closes the file. It creates and fills 10 different files in this manner. The files are given the names ATDISK.¯¯n, where *n* is a number between 0 and 9. During the second phase, the test reopens each file, reads the entire file in 2KB increments, and deletes the file. The test displays the time required to perform both phases.

The system performance measured by this final ATDISK test will vary significantly according to the number of I/O buffers used and where on the disk the test files are located. To allow meaningful comparisons to be made among various computers, the test will be run with the default number of I/O buffers (no BUFFERS= statement will be included in the CONFIG.SYS file) and a freshly formatted hard disk.

To activate this disk system performance test, which actually creates and deletes files on the hard disk, command line ATDISK –d should be used instead of just ATDISK. This ensures that if the hard disk already contains files called ATDISK.¯¯n, those files will not be deleted accidentally.

The Microsoft C compiler and assembler are used to produce the object code of the ATDISK program. Assuming that ATDISK.C contains the C source code and PDISKIO.ASM contains the assembly language source code, the following commands can be used to compile and link the program:

```
msc atdisk.c,;
masm pdiskio.asm/r,,;
link atdisk+pdiskio,,,
   slibfp.lib+slibc.lib+em.lib;
```

Although ATDISK was written expressly to test AT-compatible computers, it does not include any AT-specific code. It will run on any AT- or XT-compatible computer.

UPCOMING REVIEWS

A review of the Compaq Deskpro 286, which follows this introduction on p. 80, inaugurates this series on AT compatibles. Future articles will examine other compatibles that *PC Tech Journal* considers significant. This series will provide the facts needed to choose an AT-compatible computer based on performance and compatibility. It will give insight to those pleasant extras and unfortunate design mistakes that can characterize these AT clones.

Steven Armbrust is a freelance technical writer. Ted Forgeron works for Intel Scientific Computers as a software project manager. They recently completed writing the Programmer's Reference Manual for IBM Personal Computers, *to be published by Dow Jones-Irwin in October 1986. Paul Pierce is a microcomputer software consultant who also collects and repairs old mainframe computers in his spare time.*

LISTING 1: ATBIOS.PAS

```
{ ATBIOS -- PC Tech Journal AT BIOS Information Display Program  }
{ Version 1.00                                                    }
{ Last modified 05/23/86                                          }
{ Copyright (c) 1986, PC Tech Journal                             }
{ Program by: Paul Pierce, Ted Forgeron, Steven Armbrust          }
{ Displays pertinent information from the BIOS code and data      }
{ areas.                                                          }
{ This program is written in Turbo Pascal.  However, it can       }
{ be easily ported to any Pascal compiler that allows absolute    }
{ addressing.                                                     }

PROGRAM at_bios_info ;
  CONST
    at_id = $fc ;
    printer_mask = $c000 ;
    game_mask = $1000 ;
    serial_mask = $0e00 ;
    dma_mask = $0100 ;
    drive_num_mask = $00c0 ;
    video_mask = $0030 ;
    ndp_mask = $0002 ;
    drive_mask = $0001 ;
    co40 = $0010 ;
    co80 = $0020 ;
    mono = $0030 ;
  VAR
    i : integer ;
    romdate : ARRAY [1..9] OF char absolute $f000:$fff5 ;
    machine_id : byte absolute $f000:$fffe ;
    copyright : ARRAY [1..80] OF char absolute $f000:$e000 ;
    equip_flag : integer absolute $40:$10 ;
    mem_size : integer absolute $40:$13 ;
    key_buf : ARRAY [1..32] OF char absolute $40:$1e ;
    video_mode : byte absolute $40:$49 ;
  BEGIN
    clrscr;
    write('ATBIOS -- PC Tech Journal AT BIOS Information ') ;
    writeln('Display');
    writeln('Version 1.00, Copyright (c) 1986 PC Tech Journal') ;
    writeln;
    writeln('ROM BIOS date is                    ',romdate) ;
    IF machine_id = at_id THEN
      writeln('Machine ID is                     AT COMPATIBLE')
    ELSE
      writeln('Machine ID is                     NOT AT') ;
    write('Copyright Statement is ') ;
    window(40,6,80,7);
    gotoxy(1,1);
    write(copyright) ;
    window(1,1,80,25) ;
    gotoxy(1,8);
    write('Diskette Drives Installed             ') ;
    if (equip_flag and drive_mask) <> 0
    then writeln(((drive_num_mask and equip_flag) div 64) + 1)
    else writeln('0');
    write('80287 Math Coprocessor                ') ;
    if (equip_flag and ndp_mask) <> 0
    then writeln('YES')
    else writeln('NO');
    write('Initial Video Mode                    ') ;
    case (equip_flag and video_mask) of
      co40: writeln('CGA 40x25 B/W text');
      co80: writeln('CGA 80x25 B/W text');
      mono: writeln('Monochrome 80x25 text');
    end;
    write('Current Video Mode                    ') ;
    case video_mode of
      00: writeln('CGA 40x25 B/W text');
      01: writeln('CGA 40x25 16-color text');
      02: writeln('CGA 80x25 B/W text');
      03: writeln('CGA 80x25 16-color text');
      04: writeln('CGA 320x200 4-color graphics');
      05: writeln('CGA 320x200 4-gray graphics');
```

```
    06: writeln('CGA 640x200 B/W graphics');
    07: writeln('Monochrome 80x25 text');
    08: writeln('JR 160x200 16-color graphics');
    09: writeln('JR 320x200 16-color graphics');
    10: writeln('EGA 640x200 4/64-color graphics');
    13: writeln('EGA 320x200 16-color graphics');
    14: writeln('EGA 640x200 16-color graphics');
    15: writeln('EGA 640x350 4-color graphics');
  end;
  write('DMA Present                          ');
  if (equip_flag and dma_mask) <> 0
  then writeln('NO')
  else writeln('YES');
  writeln('RS-232 Serial Ports                     ',
        (equip_flag and serial_mask) div 512);
  write('Game Adapter Present                 ');
  if (equip_flag and game_mask) <> 0
  then writeln('YES')
  else writeln('NO');
  writeln('Parallel Printer Ports                ',
        abs((equip_flag and printer_mask) div 16384));
  writeln('Memory Size in K Bytes                ',mem_size);
  write('Keyboard Buffer Contents             ');
  i := 1;
  while i <= 32 do
  begin
    if (key_buf[i] < '!') or (key_buf > ')') then
      write('.')
    else write(key_buf[i]);
    i := i + 2 ;
  end;
  writeln;
END.
```

LISTING 2: ATKEY.ASM

```
; ATKEY -- PC Tech Journal AT Keyboard Compatibility Test
; Version 1.00
; Last modified 05/23/86
; Copyright (c) 1986, PC Tech Journal
; Program by: Paul Pierce, Ted Forgeron, Steven Armbrust
; Checks for AT keyboard compatibility by trying to speed
; the keyboard typematic rate to 30 reps per second and a
; keyboard delay of 1/4 second.
; See AT Technical Reference Manual pages 4-5 thru 4-8 for
; more information on programming the AT keyboard.

code    segment public

assume cs:code,ds:code

        org     100h                    ;set up as a COM file
start:  jmp     begin

msg     db 0dh,0ah
        db 'ATKEY -- PC Tech Journal AT Keyboard Compatibility '
        db 'Test',0dh,0ah
        db 'Version 1.00, Copyright (c) 1986 PC Tech Journal',
        db 0dh,0ah,0ah,'$'
badmsg  db 'Keyboard is incompatible with IBM PC/AT keyboard',
        db 0dh,0ah,'$'
goodmsg db 'Keyboard is compatible with IBM PC/AT keyboard',
        db 0dh,0ah,0dh,0ah
        db 'Keyboard will now repeat at 30 characters per second',
        db 0dh,0ah,'with 1/4 second delay.',0dh,0ah,'$'

begin:  mov     ax,cs                   ;set up ds
        mov     ds,ax                   ;to same seg as cs
        mov     dx,offset msg           ;print msg on screen
        mov     ah,09h                  ;dos print string funct
        int     21h                     ;call dos
        mov     al,0f3h                 ;typematic rate cmd
        mov     dx,60h                  ;8042 scan/diag port
        mov     cx,2000                 ;ack retry count
        out     dx,al                   ;send command to 8042
ack:    in      al,60h                  ;in from 8042
        dec     cx                      ;decrement retry count
        jnz     cont                    ;try again
        mov     dx,offset badmsg        ;print msg on screen
        mov     ah,09h                  ;dos print string funct
        int     21h                     ;call dos
        jmp     done                    ;give up and terminate
cont:   cmp     al,0fah                 ;is it an ack?
        jne     ack                     ;no, try again
        mov     al,0                    ;set rate to code for 30
        out     dx,al                   ;output command to 8042
        mov     dx,offset goodmsg       ;print msg on screen
        mov     ah,09h                  ;dos print string funct
        int     21h                     ;call dos
done:   mov     ah,4ch                  ;dos terminate process
        int     21h                     ;call dos
code    ends                            ;end of code segement
        end     start                   ;start is the entry point
```

LISTING 3: ATPERF.C

```
/*
 * ATPERF -- PC Tech Journal AT Hardware Performance Test
 * Version 1.00
 * Last modified 05/17/86
 * Copyright (c) 1986, PC Tech Journal
 * Program by: Paul Pierce, Ted Forgeron, Steven Armbrust
 * Measures clock rates and memory speeds
 * of AT compatible computers.
 */
#define vars 14

/* Timer rate in MHz */
#define TIMER2_RATE 1.193180
/* Number of processor clocks in a multiply instruction */
#define MULCLKS 21
/* Overhead in the multiply test */
#define MULOVH ( 15 + 14*count/100 )
/* Overhead in the mov instruction test */
#define MOVOVH ( clktime * (15 + 14*count/100) )
/* Overhead in the stringop tests */
#define STROVH ( clktime * 8 )
/* Number of numeric processor clocks in a FP divide */
#define FPCLKS 203
/* Processor overhead in the FP divide test */
#define FPOVH ( clktime * 9 * FPCOUNT )
/* Count for most tests */
#define COUNT 1000
/* Count for the f. p. divide test */
#define FPCOUNT 100
/* Number of trials for each test */
#define TRIALS 100

double clkrate; /* Processor clock rate, MHz */
double clktime; /* Processor clock period, usec */
double fprate;  /* FP processor clock rate, MHz */
double fpacc;   /* FP processor clock period accumulator */
int emmok;      /* Set if extended memory is present */
int ndpok;      /* Set if math coprocessor is present */

/*
 * Main program.
 */
main(argc, argv)
        int     argc;
        char    **argv;
{
        double raw, brw, wrw;   /* Variables for raw data */
        double acctime[vars];   /* Accumulators for speeds */
        int count;              /* Number of ops per trial */
        int trials;             /* Number of repetitions */
        register int i;

        count = COUNT;
        trials = TRIALS;
        /*
         * Measure the clock rate by executing
         * multiply instructions.  Each multiply
         * takes a fixed number of clock cycles.
         */
        clktime = 0;
        for (i = 0; i < trials; i++) {
                /*
                 * Obtain the number of clock ticks for
                 * "count" multiplies.
                 */
                raw = multime(count);
```

```
        /*
         * Accumulate the clock time in microseconds
         * by adjusting for the timer rate,
         * number of clocks per multiply,
         * instruction count, and test overhead.
         */
        clktime +=  raw /
                (TIMER2_RATE * (MULCLKS*count + MULOVH));
}
/*
 * Calculate the average clock period by dividing by
 * the number of trials.  The clock rate is the
 * inverse of the clock period.
 */
clktime /= trials;
clkrate = 1.0/clktime;
/*
 * Determine whether there a math coprocessor
 * in the system.
 */
ndpok = ndp_present();
/*
 * Determine whether there is extended memory in the
 * system and allocate a piece of it for testing.
 */
emmok = (setup_emm() == 0);

/*
 * Clear all of the memory speed accumulators.
 */
for (i = 0; i < vars; i++)
        acctime[i] = 0;
/*
 * Do the memory speed tests.
 */
for (i = 0; i < trials; i++) {
        /*
         * Obtain the number of timer ticks for
         * "count" mov instructions, which are
         * limited by memory fetch time.
 */
raw = wmovtime(count);
/*
 * Accumulate the number of microseconds
 * per instruction fetch by adjusting for
 * the timer rate, test overhead, and
 * instruction count.
 */
acctime[8] +=
        (raw / TIMER2_RATE - MOVOVH) / count;
/*
 * In a similar manner, measure the
 * RAM byte write time using the stos
 * instruction.
 */
brw = bstotime(count);
acctime[2] += brw/(TIMER2_RATE*count);
/*
 * Calculate memory read time by
 * measuring movs instructions
 * (read followed by write) and
 * subtracting the write time
 * measured above.
 */
raw = bmvstime(count) - brw;
acctime[0] += raw/(TIMER2_RATE*count);
/*
 * Calculate ROM read time by
 * measuring movs from ROM to RAM
 * and subtracting RAM write time
 * as above.
 */
raw = bromtime(count) - brw;
acctime[4] += raw/(TIMER2_RATE*count);
/*
 * Make similar measurements for
 * word operations.
 */
wrw = wstotime(count);
acctime[3] += raw/(TIMER2_RATE*count);
raw = wmvstime(count) - wrw;
acctime[1] += raw/(TIMER2_RATE*count);
raw = wromtime(count) - wrw;
acctime[5] += raw/(TIMER2_RATE*count);
/*
 * If EMM is present, do measurements
 * on it using the same techniques.
 */
if (emmok) {
        /*
         * Measure byte write,
         * calculate byte read.
         */
                brw = bemstime(count);
                acctime[12] += brw/(TIMER2_RATE*count);
                raw = bemmtime(count) - brw;
                acctime[10] += raw/(TIMER2_RATE*count);
                /*
                 * Measure word write,
                 * calculate word read.
                 */
                wrw = wemstime(count);
                acctime[13] += raw/(TIMER2_RATE*count);
                raw = wemmtime(count) - wrw;
                acctime[11] += raw/(TIMER2_RATE*count);
        }
        /*
         * Measure byte and word writes
         * into video RAM (assuming CGA adapter.)
         */
        raw = bvidtime(count);
        acctime[6] += raw/(TIMER2_RATE*count);
        raw = wvidtime(count);
        acctime[7] += raw/(TIMER2_RATE*count);
}
/*
 * Calculate averages for all measurements.
 */
for (i = 0; i < vars; i++)
        acctime[i] /= trials;
/*
 * Release EMM memory page.
 */
if (emmok)
        finish_emm();
/*
 * Calculate numeric processor clock
 * rate using floating point divide
 * instructions, using the same
 * technique as was used to measure
 * the processor clock rate.
 */
if (ndpok) {
        fprate = 0;
        for (i = 0; i < trials; i++) {
                raw = fptime(FPCOUNT);
                fpacc +=  (raw / TIMER2_RATE - FPOVH) /
                        FPCLKS / FPCOUNT;
        }
        fpacc /= trials;
        fprate = 1.0/fpacc;
}
/*
 * Display the basic measurement results and
 * performance index relative to a 6 MHz AT.
 */
printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
printf("ATPERF -- PC Tech Journal AT Hardware ");
printf("Performance Test\n");
printf("Version 1.00, Copyright (c) 1986 PC Tech ");
printf("Journal\n");
printf("IBM PC/AT model 339 (8 MHz) = 1.00 for relative ");
printf("measurements.\n                              ");
printf("      Byte           Word       Relative\n");

printf("Average RAM instr. fetch:   ");
printf("                    ");
printf("%10.3g uS", acctime[8]);
printf("%10.2g\n", 0.403/acctime[8]);

printf("Average RAM read time:      ");
printf("%10.3g uS    ", acctime[0]);
```

```
printf("%10.3g uS", acctime[1]);
printf("%10.2g\n", 0.401/acctime[1]);

printf("Average RAM write time:     ");
printf("%10.3g uS    ", acctime[2]);
printf("%10.3g uS", acctime[3]);
printf("%10.2g\n", 0.401/acctime[3]);

if (emmok) {
        printf("Average EMM read time:      ");
        printf("%10.3g uS    ", acctime[10]);
        printf("%10.3g uS", acctime[11]);
        printf("%10.2g\n", 0.402/acctime[11]);

        printf("Average EMM write time:     ");
        printf("%10.3g uS    ", acctime[12]);
        printf("%10.3g uS", acctime[13]);
        printf("%10.2g\n", 0.402/acctime[13]);
}
printf("Average ROM read time:      ");
printf("%10.3g uS    ", acctime[4]);
printf("%10.3g uS", acctime[5]);
printf("%10.2g\n", 0.401/acctime[5]);

printf("Average Video write time:   ");
printf("%10.3g uS    ", acctime[6]);
printf("%10.3g uS", acctime[7]);
printf("%10.2g\n", 2.415/acctime[7]);

printf("\nClock rate:                    ");
printf("%4.1g MHz", clkrate);
printf("  Relative: %4.2g\n", clkrate/8.0);

if (ndpok) {
        printf("Numeric processor clock rate: ");
        printf("%4.1g MHz", fprate);
        printf("  Relative: %4.2g\n", fprate/5.33);

        }
        /*
         * Calculate refresh overhead from instruction
         * fetch time by assuming that each fetch takes
         * an exact multiple of the clock period.  The
         * difference between average time and the time
         * for an individual fetch is due to memory
         * refresh cycles.
         */
        raw = acctime[0] / clktime;
        printf("Refresh overhead:              %2.1g%%\n",
                ( (raw - (int)raw) / (int)raw ) * 100);
        /*
         * Print information about the memory based
         * on the speed measurements.
         */
        printf("\nMemory   ");
        printf("       Access width          Wait states\n");
        analyze("RAM read", acctime[0], acctime[1]);
        analyze("RAM write", acctime[2], acctime[3]);
        if (emmok) {
                analyze("EMM read", acctime[10], acctime[11]);
                analyze("EMM write", acctime[12], acctime[13]);
        }
        analyze("ROM read", acctime[4], acctime[5]);
        analyze("Video write", acctime[6], acctime[7]);
}
/*
 * analyze
 * This procedure deduces information about the memory based on
 * the measured times.
 * If byte (8 bits) and word (16 bits) times are different then
 * the memory is byte oriented since each word operation takes
 * two byte operations.  Otherwise, if the byte and word
 * times are about the same, the memory is word oriented and can
 * access either a word or a byte in a single memory cycle.
 *
 * Each memory access takes an exact number of processor clock
 * cycles.  The first two are required by the processor, but
 * any additional cycles are determined by the memory and are
 * called wait states (because the processor is waiting for
 * the memory.)
 */
analyze(name, btime, wtime)
        char   *name;
        double  btime;
        double  wtime;
{
        /*
         * Print the heading
         */
        printf("%-12s", name);
        /*
         * Determine whether the memory is byte
         * oriented, word oriented, or neither.
         * (If neither, the data are suspect.)
         */
        if (btime > wtime*0.75 &&
            btime < wtime*1.25)
                printf("       Word  ");
        else if (btime*2 > wtime*0.75 &&
                 btime*2 < wtime*1.25)
                printf("       Byte  ");
        else
                printf("      Strange");
        /*
         * Determine the number of wait states
         * by subtracting two processor clock times,
         * dividing by the clock period,
         * and rounding down to an integer.
         */
        printf("             %6.0f\n",
                (btime - 2*clktime) / clktime);
}
```

LISTING 4: TIME.ASM

```
                NAME    TIME

_TEXT   SEGMENT  BYTE PUBLIC 'CODE'
_TEXT   ENDS
CONST   SEGMENT  WORD PUBLIC 'CONST'
CONST   ENDS
_BSS    SEGMENT  WORD PUBLIC 'BSS'
_BSS    ENDS
_DATA   SEGMENT  WORD PUBLIC 'DATA'
_DATA   ENDS
DGROUP  GROUP   CONST,  _BSS,   _DATA
        ASSUME  CS: _TEXT, DS: DGROUP, SS: DGROUP, ES: DGROUP

TESTSEG SEGMENT WORD PUBLIC 'TEST'

TESTSEG_START   DW      32767 DUP (?)

TESTSEG ENDS

PPI_PORT        EQU     061H
TIMER2_PORT     EQU     042H
TIMER_CTRL      EQU     043H

_DATA           SEGMENT

EMMBASE         DW      9000H
PID             DW      ?
EMM_NAME        DB      "EMMXXXX0"

_DATA           ENDS

_TEXT           SEGMENT

;*****************************************************************;
;                                                                 ;
;       _MULTIME                                                  ;
;       TIME EXECUTION OF MULTIPLY INSTRUCTIONS                   ;
;                                                                 ;
;*****************************************************************;

                PUBLIC  _MULTIME
_MULTIME        PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
```

```
        MOV     DI, 0                   ; CLEAR DI
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        ADD     AX, 99                  ; ROUND UP
        MOV     CX, 100                 ; DIVIDE BY 100 =
        DIV     CL                      ;  NUMBER OF INSTRUCTIONS
        MOV     CL, AL                  ;  PER PASS
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
ML:     MUL     DI                      ; DO 100 MULTIPLIES
        MUL     DI                      ;
        ...
<<<<<The above line is repeated a total of 99 times>>>>>

        DEC     CX                      ; COUNT THIS PASS
        JZ      MD                      ; JUMP IF COMPLETE
        JMP     ML                      ; LOOP BACK IF NOT DONE
MD:     MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_MULTIME        ENDP

;*******************************************************************;
;                                                                   ;
;       _WMOVTIME                                                   ;
;       TIME EXECUTION OF MOV INSTRUCTION (INSTR. READ TIME)        ;
;                                                                   ;
;*******************************************************************;

                PUBLIC  _WMOVTIME
_WMOVTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, 0                   ; CLEAR DI
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        ADD     AX, 99                  ; ROUND UP
        MOV     CX, 100                 ; DIVIDE BY 100 =
        DIV     CL                      ;  NUMBER OF INSTRUCTIONS
        MOV     CL, AL                  ;  PER PASS
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
IL:     MOV     DX, BX                  ; DO 100 MULTIPLIES
        MOV     DX, BX                  ;
        ...
<<<<<The above line is repeated a total of 99 times>>>>>
        DEC     CX                      ; COUNT THIS PASS
        JZ      ID                      ; JUMP IF COMPLETE
        JMP     IL                      ; LOOP BACK IF NOT DONE
ID:     MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_WMOVTIME       ENDP

;*******************************************************************;
;                                                                   ;
;       _BMVSTIME                                                   ;
;       TIME EXECUTION OF REP MOVSB INSTRUCTION                     ;
;                                                                   ;
;*******************************************************************;

                PUBLIC  _BMVSTIME
_BMVSTIME       PROC    NEAR
```

```
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        LEA     SI, TESTSEG_START       ; DS:SI -> TEST SEGMENT
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_BMVSTIME       ENDP

;*******************************************************************;
;                                                                   ;
;       _WMVSTIME                                                   ;
;       TIME EXECUTION OF REP MOVSW INSTRUCTION                     ;
;                                                                   ;
;*******************************************************************;

                PUBLIC  _WMVSTIME
_WMVSTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        LEA     SI, TESTSEG_START       ; DS:SI -> TEST SEGMENT
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_WMVSTIME       ENDP

;*******************************************************************;
;                                                                   ;
;       _BROMTIME                                                   ;
;       TIME EXECUTION OF REP MOVSB INSTRUCTION FROM ROM            ;
;                                                                   ;
;*******************************************************************;
```

```
                PUBLIC  _BROMTIME
_BROMTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DI, 0F000H              ; SET DS TO ROM START
        MOV     DS, DI                  ;
        MOV     SI, 0                   ; DS:SI -> ROM
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_BROMTIME       ENDP

;*****************************************************************;
;                                                                 ;
;       _WROMTIME                                                 ;
;       TIME EXECUTION OF REP MOVSW INSTRUCTION FROM ROM          ;
;                                                                 ;
;*****************************************************************;

                PUBLIC  _WROMTIME
_WROMTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DI, 0F000H              ; SET DS TO ROM START
        MOV     DS, DI                  ;
        MOV     SI, 0                   ; DS:SI -> ROM
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_WROMTIME       ENDP
```

```
;*****************************************************************;
;                                                                 ;
;       _BSTOTIME                                                 ;
;       TIME EXECUTION OF REP STOSB INSTRUCTION                   ;
;                                                                 ;
;*****************************************************************;

                PUBLIC  _BSTOTIME
_BSTOTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    ES                      ; SAVE ES
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP STOSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     ES                      ; RESTORE ES
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_BSTOTIME       ENDP

;*****************************************************************;
;                                                                 ;
;       _WSTOTIME                                                 ;
;       TIME EXECUTION OF REP STOSW INSTRUCTION                   ;
;                                                                 ;
;*****************************************************************;

                PUBLIC  _WSTOTIME
_WSTOTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    ES                      ; SAVE ES
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP STOSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     ES                      ; RESTORE ES
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_WSTOTIME       ENDP

;*****************************************************************;
;                                                                 ;
;       _BVIDTIME                                                 ;
;       TIME EXECUTION OF REP STOSB INTO VIDEO MEMORY             ;
;                                                                 ;
;*****************************************************************;

                PUBLIC  _BVIDTIME
```

```
_BVIDTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    ES                      ; SAVE ES
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, 0B800H              ;
        MOV     ES, DI                  ;
        MOV     DI, 0                   ; ES:DI -> VIDEO MEMORY
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP STOSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     ES                      ; RESTORE ES
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_BVIDTIME       ENDP


;******************************************************************;
;                                                                  ;
;       _WVIDTIME                                                  ;
;       TIME EXECUTION OF REP STOSW INTO VIDEO MEMORY              ;
;                                                                  ;
;******************************************************************;

                PUBLIC  _WVIDTIME
_WVIDTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    ES                      ; SAVE ES
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, 0B800H              ;
        MOV     ES, DI                  ;
        MOV     DI, 0                   ; ES:DI -> VIDEO MEMORY
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP STOSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     ES                      ; RESTORE ES
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_WVIDTIME       ENDP

;******************************************************************;
;                                                                  ;
;       _FPTIME                                                    ;
;       TIME EXECUTION OF FLOATING POINT DIVIDE                    ;
;                                                                  ;
;******************************************************************;

                PUBLIC  _FPTIME
_FPTIME         PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
```

```
        MOV     DI, 0                   ; CLEAR DI
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        ADD     AX, 99                  ; ROUND UP
        MOV     CX, 100                 ; DIVIDE BY 100 =
        DIV     CL                      ;  NUMBER OF INSTRUCTIONS
        MOV     CL, AL                  ;  PER PASS
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        FNINIT                          ; INIT FP
        FLD1                            ; DIVIDE 1.0
        FLD1                            ; BY 1.0
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
FL:     FDIV    ST(1), ST               ; DO 100 DIVIDES
        FDIV    ST(1), ST               ;
        ...
<<<<<The above line is repeated a total of 99 times>>>>>

        DEC     CX                      ; COUNT THIS PASS
        JZ      FD                      ; JUMP IF COMPLETE
        JMP     FL                      ; LOOP BACK IF NOT DONE
FD:     MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_FPTIME         ENDP

;******************************************************************;
;                                                                  ;
;       _BEMSTIME                                                  ;
;       TIME EXECUTION OF REP STOSB INSTRUCTION                    ;
;                                                                  ;
;******************************************************************;

                PUBLIC  _BEMSTIME
_BEMSTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    ES                      ; SAVE ES
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     ES, EMMBASE             ; SET UP EMM BASE ADDRESS
        XOR     DI, DI                  ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP STOSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     ES                      ; RESTORE ES
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_BEMSTIME       ENDP

;******************************************************************;
;                                                                  ;
;       _WEMSTIME                                                  ;
;       TIME EXECUTION OF REP STOSW INSTRUCTION                    ;
;                                                                  ;
;******************************************************************;

                PUBLIC  _WEMSTIME
_WEMSTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    ES                      ; SAVE ES
```

```
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     ES, EMMBASE             ; SET UP EMM BASE ADDRESS
        XOR     DI, DI                  ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP STOSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     ES                      ; RESTORE ES
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_WEMSTIME       ENDP
;***************************************************************;
;                                                               ;
;       _BEMMTIME                                               ;
;       TIME EXECUTION OF REP MOVSB INSTRUCTION                 ;
;                                                               ;
;***************************************************************;

                PUBLIC  _BEMMTIME
_BEMMTIME       PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, EMMBASE             ; SET UP EMM BASE ADDRESS
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        XOR     DI, DI                  ; ES:DI -> TEST SEGMENT
        XOR     SI, SI                  ; DS:SI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_BEMMTIME       ENDP

;***************************************************************;
;                                                               ;
;       _WEMMTIME                                               ;
;       TIME EXECUTION OF REP MOVSW INSTRUCTION                 ;
;                                                               ;
;***************************************************************;

                PUBLIC  _WEMMTIME
_WEMMTIME       PROC    NEAR
                PUSH    BP                      ; SAVE FRAME
                MOV     BP, SP                  ;
                PUSH    DS                      ; SAVE DS
                PUSH    ES                      ; SAVE ES
                PUSH    SI                      ; SAVE SI
                PUSH    DI                      ; SAVE DI
                CALL    SETUP_TIMER             ; SET UP TIMER
```

```
        MOV     DI, EMMBASE             ; SET UP EMM BASE ADDRESS
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        XOR     DI, DI                  ; ES:DI -> TEST SEGMENT
        XOR     SI, SI                  ; DS:SI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_WEMMTIME       ENDP
;***************************************************************;
;                                                               ;
;       _SETUP_EMM                                              ;
;       SET UP EXPANDED MEMORY AND RETURN THE BASE              ;
;                                                               ;
;***************************************************************;

                PUBLIC  _SETUP_EMM
_SETUP_EMM      PROC    NEAR

        PUSH    BP                      ; SAVE REGISTERS
        PUSH    ES                      ;
        PUSH    SI                      ;
        PUSH    DI                      ;
        MOV     AH, 35H                 ; GET EMM INTERRUPT
        MOV     AL, 67H                 ; VECTOR
        INT     21H                     ;
        MOV     AX, ES                  ; CHECK FOR VALID BASE
        MOV     BX, CS                  ; MUST BE BELOW CS
        CMP     AX, BX                  ;
        JA      SENO                    ; JUMP IF NOT GOOD
        MOV     DI, 000AH               ; OFFSET OF DRIVER NAME
        LEA     SI, EMM_NAME            ; COMPARE STRING
        MOV     CX, 8                   ; LENGTH OF STRING
        CLD
        REPE    CMPSB                   ; COMPARE THE NAME
        JNE     SENO                    ; JUMP IF NO GOOD
                                        ;
SE1:    MOV     AH, 40H                 ; FUNCTION 1:
        INT     67H                     ; GET MANAGER STATUS
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE1                     ; TRY AGAIN IF BUSY
        OR      AH, AH                  ; CHECK FOR ERROR
        JNZ     SENO                    ; JUMP ON ERROR
                                        ;
SE2:    MOV     AH, 41H                 ; FUNCTION 2:
        INT     67H                     ; GET PAGE FRAME BASE
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE2                     ; TRY AGAIN IF BUSY
        OR      AH, AH                  ; CHECK FOR ERROR
        JNZ     SENO                    ; JUMP ON ERROR
        MOV     EMMBASE, BX             ; SAVE THE BASE
                                        ;
SE3:    MOV     AH, 42H                 ; FUNCTION 3:
        INT     67H                     ; GET NUMBER OF PAGES
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE3                     ; TRY AGAIN IF BUSY
        OR      AH, AH                  ; CHECK FOR ERROR
        JNZ     SENO                    ; JUMP ON ERROR
        OR      BX, BX                  ; CHECK UNALLOCATED PAGES
        JZ      SENO                    ; JUMP IF NONE AVAILABLE
                                        ;
SE4:    MOV     AH, 43H                 ; FUNCTION 4:
        MOV     BX, 1                   ; ALLOCATE ONE PAGE
        INT     67H                     ;
        CMP     AH, 82H                 ; CHECK FOR BUSY
```

```
                SE4                     ; TRY AGAIN IF BUSY
                AH, AH                  ; CHECK FOR ERROR
                SENO                    ; JUMP ON ERROR
        OV      PID, DX                 ; SAVE THE PROCESS ID
                                        ;
        MOV     AH, 44H                 ; FUNCTION 5:
        XOR     BX, BX                  ; MAP THE PAGE TO
        XOR     AL, AL                  ;  FRAME BASE
        INT     67H                     ;
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE5                     ; TRY AGAIN IF BUSY
        OR      AH, AH                  ; CHECK FOR ERROR
        JNZ     SENC                    ; JUMP ON ERROR
                                        ;
        XOR     AX, AX                  ;
        POP     DI                      ; RESTORE REGISTERS
        POP     SI                      ;
        POP     ES                      ;
        POP     BP                      ;
        RET                             ; RETURN 0

SENC:   MOV     AH, 45H                 ; FUNCTION 6:
        INT     67H                     ; CLOSE EMM
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SENC                    ; TRY AGAIN IF BUSY
SENO:   MOV     AX, 0FFFFH              ;
        POP     DI                      ; RESTORE REGISTERS
        POP     SI                      ;
        POP     ES                      ;
        POP     BP                      ;
        RET                             ; RETURN -1

_SETUP_EMM      ENDP

;*****************************************************************;
;                                                                 ;
;       _FINISH_EMM                                               ;
;       CLOSE THE EMM DEVICE, RELEASE THE PAGE                    ;
;                                                                 ;
;*****************************************************************;
                PUBLIC  _FINISH_EMM
_FINISH_EMM     PROC    NEAR

        PUSH    BP                      ; SAVE REGISTERS
        PUSH    ES                      ;
        PUSH    SI                      ;
        PUSH    DI                      ;
SE6:    MOV     AH, 45H                 ; FUNCTION 6:
        MOV     DX, PID                 ; CLOSE EMM
        INT     67H                     ;
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE6                     ; TRY AGAIN IF BUSY
        POP     DI                      ; RESTORE REGISTERS
        POP     SI                      ;
        POP     ES                      ;
        POP     BP                      ;
        RET                             ; RETURN

_FINISH_EMM     ENDP


;*****************************************************************;
;                                                                 ;
;       SETUP_TIMER                                               ;
;       SET UP THE TIMER FOR MAXIMUM COUNT, TO TIME A RUN         ;
;                                                                 ;
;*****************************************************************;

SETUP_TIMER     PROC    NEAR

        PUSH    AX                      ; SAVE AX
        IN      AL, PPI_PORT            ; STOP THE TIMER
        AND     AL, 0FCH                ;
        OUT     PPI_PORT, AL            ;
        MOV     AL, 0B4H                ; INITIALIZE THE TIMER
        OUT     TIMER_CTRL, AL          ;
        MOV     AL, 0                   ; CLEAR THE COUNT
        OUT     TIMER2_PORT, AL         ;
        NOP                             ;
        OUT     TIMER2_PORT, AL         ;
        POP     AX                      ; RESTORE AX
        RET                             ; RETURN

SETUP_TIMER     ENDP


;*****************************************************************;
;                                                                 ;
;       GET_TIMER                                                 ;
;       TAKE THE COUNT FROM THE TIMER                             ;
;                                                                 ;
;*****************************************************************;
GET_TIMER       PROC    NEAR

        PUSH    BX                      ; SAVE REGISTERS
        IN      AL, TIMER2_PORT         ; GET LOW BYTE OF TIME
        MOV     AH, AL                  ;
        IN      AL, TIMER2_PORT         ; GET HIGH BYTE
        XCHG    AL, AH                  ; TIME IN AX
        NEG     AX                      ; CORRECT FOR COUNT-DOWN
        POP     BX                      ; RESTORE REGISTERS
        RET                             ; RETURN

GET_TIMER       ENDP

;*****************************************************************;
;                                                                 ;
;       _NDP_PRESENT                                              ;
;       CHECK IF 80287 IS PRESENT                                 ;
;                                                                 ;
;*****************************************************************;
                PUBLIC  _NDP_PRESENT
_NDP_PRESENT    PROC    NEAR

        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        INT     11H                     ; BIOS EQUIP CHECK
        TEST    AL,02H                  ; IS 80287 BIT SET?
        JZ      NO                      ; NO MEANS NO 80287
        MOV     AX,01h                  ; RETURN TRUE
        JMP     EXIT                    ; ALL DONE
NO:     XOR     AX,AX                   ; SET AX TO FALSE
EXIT:   MOV     SP, BP                  ; RESTORE SP
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN

_NDP_PRESENT    ENDP

_TEXT           ENDS

                END
```

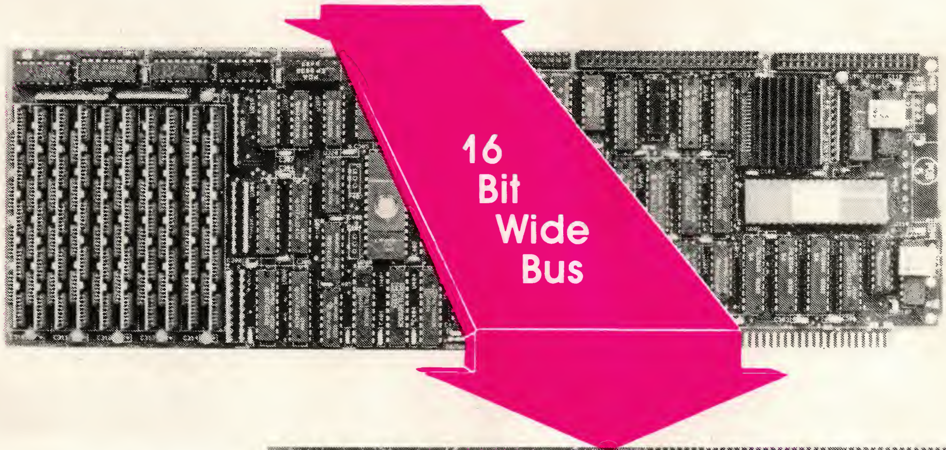
LISTING 5: ATFLOAT.C

```
/*
 * ATFLOAT -- PC Tech Journal Floating-Point Performance Test
 * Version 1.00
 * Last modified 05/23/86
 * Copyright (c) 1986, PC Tech Journal
 * Program by: Paul Pierce, Ted Forgeron, Steven Armbrust
 * Measures the time it takes to multiply two matrices of
 * double-precision floating-point numbers and compares it
 * to the time it takes an 8MHz PC/AT with an 80287 math
 * coprocessor to do the same.
 */
/* Number of iterations the test runs for. */
#define TRIALS 100
/* Dimension of the matrix */
#define SIZE 20
unsigned long time();
unsigned rand();

double drand()
{
        return (double)rand() / 32767;
}
        double a[SIZE][SIZE];
        double b[SIZE][SIZE];

        main()
        {
        int i;
```

FREE
EXPANDED MEMORY BOARD!*

"Make no mistake, the PC-286 is the most awesome piece of high technology you can put in a PC..."
— BUSINESS COMPUTER SYSTEMS

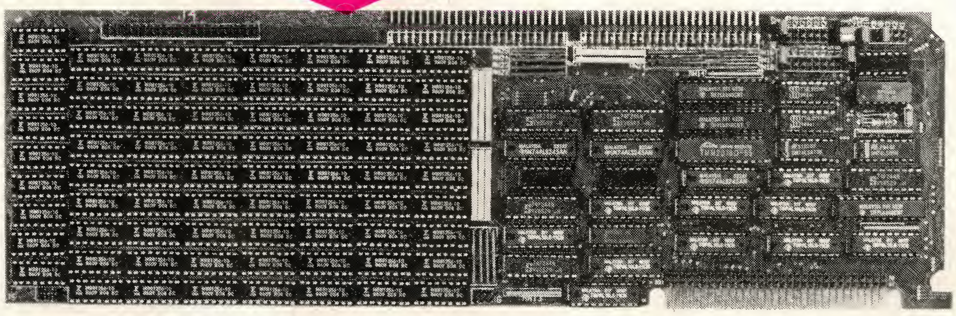
"Of this month's selections, PC-286 was fastest for the spreadsheet recalculation."
— LOTUS MAGAZINE



— 80286 (6, 8, 10, 12.5 MHz)

— Optional 80287 (5, 8, 10 MHz)

- Fully compatible with the industry-standard Lotus/Intel/Microsoft (LIM) Expanded Memory Specification (EMS), version 3.2
- Expandable up to 8 megabytes.
- Compatible with all major productivity software such as Lotus 1-2-3, Symphony, Framework II.



• BUNDLED WITH MICROSOFT WINDOWS •

Top Board processes EMS-applications up to three times faster than any other EMS board on the IBM PC XT. Top Board is the only EMS expanded memory board with both an 8 and 16 bit data bus and can function across the 16 bit wide bus with the STD-powered PC-286 card.

Top Board frees you from the memory limits of 640K. Expand memory up to 2 MB with the confidence of LIM-compatible standards. Unleash this full potential with processor speeds up to 12 MHz when used with our PC-286 board across the built-in, 16 bit data bus to the Top Board.

Expect performance that exceeds industry standards. Together, the PC-286 and Top Board perform beyond their individual state-of-the-art designs. Your CAD, spreadsheet, database or other software designed to utilize the EMS specification will run up to 8 times faster than an IBM PC XT with this product combination.

The Top Board is fully compatible with the Lotus/Intel/Microsoft Expanded Memory Specification. It's unique design has an efficient and flexible paging scheme for even greater performance with EMS-compatible applications and multitasking software.

For the performance you and your PC have been waiting for, order a Top Board today. Solve your PC's problems with processing-intensive applications and improve your own personal productivity.

CALL (206) 883-8440

STD
SEATTLE TELECOM & DATA, INC.
2637 151st Pl. N.E.
Redmond, WA 98052

Top Board and PC-286 are trademarks of STD. Framework is a trademark of Ashton-Tate. Lotus 1-2-3 and Symphony are trademarks of Lotus Development Corporation. IBM PC XT is a trademark of IBM Corp. Contact us regarding optional EGA graphics and multifunction daugther boards.

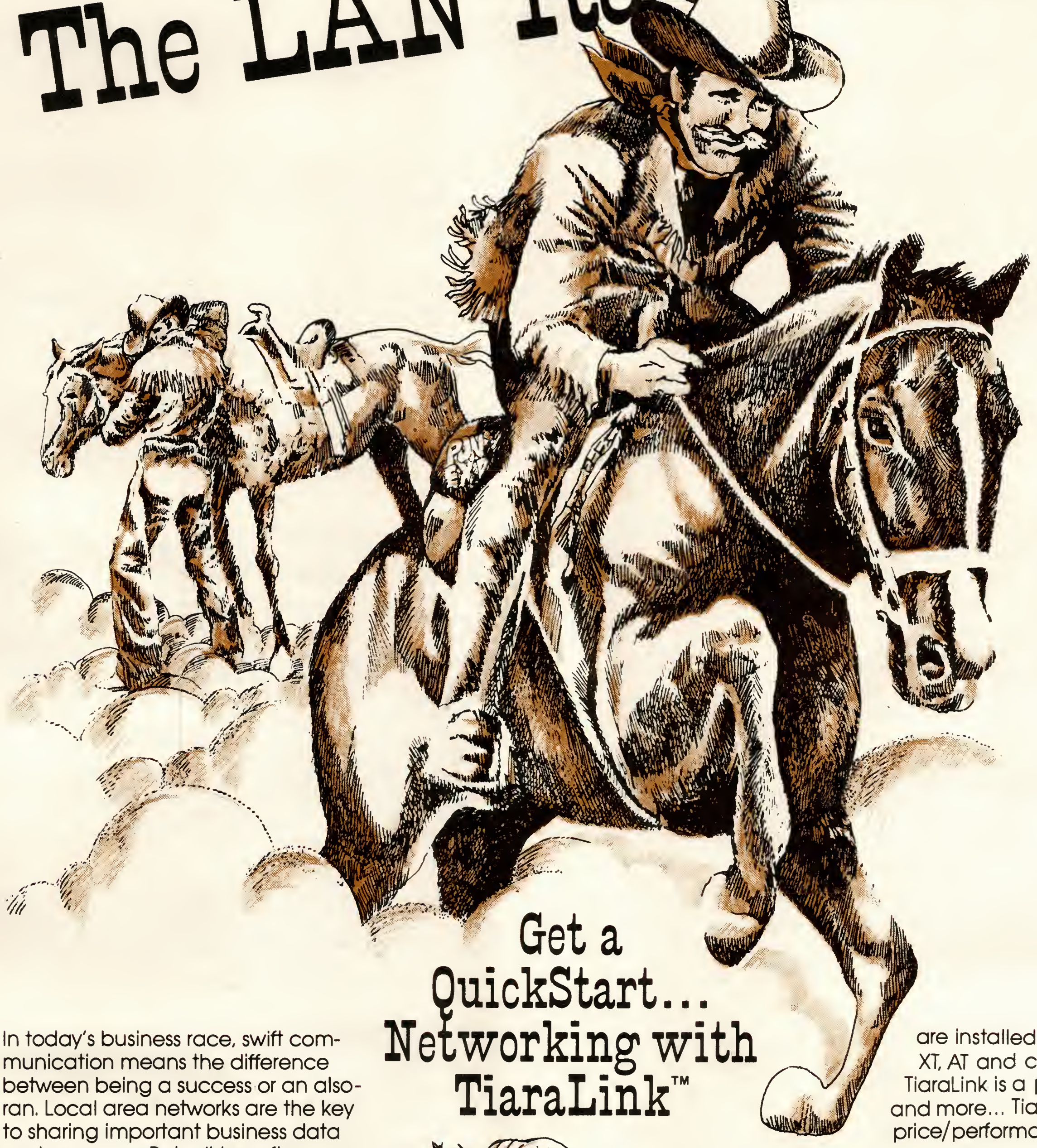
***Must purchase a 6 or 8 MHz expandable PC-286 board to receive a free 512K Top Board ($549 value). Offer expires September 30, 1986.**

```
egister j, k;
nt n;
        unsigned long start;
        unsigned long total;
        unsigned trials;
        double t;

        trials = TRIALS;
        /*
         * Fill matrix with random numbers.
         */
        for (i = 0; i < SIZE; i++)
                for (j = 0; j < SIZE; j++)
                        a[i][j] = drand();
        /*
         * Repeatedly multiply the matrices and
         * report the relative and absolute times.
         */
        start = time();
        printf("\nATFLOAT -- PC Tech Journal AT Floating-Point");
        printf(" Performance Test\n");
        printf("Version 1.00, Copyright (c) 1986 PC Tech ");
        printf("Journal\n");
        printf("\nThis test runs for %d iterations ...\n", trials);
        for (n = 1; n <= trials; n++) {
                printf("%d\r", n);
                for (i = 0; i < SIZE; i++) {
                        for (j = 0; j < SIZE; j++) {
                                t = 0;
                                for (k = 0; k < SIZE; k++)
                                        t += a[k][j] * a[i][k];
                                b[i][j] = t;
                        }
                }
        }
        total = time() - start;
        printf("\rElapsed time is %ld seconds.\n\n", total, trials);
        printf("Floating-point performance index relative\n");
        printf("to 8MHz IBM PC/AT with 80287 = %2.1f\n",
                94.0 / (float) total);
}
```

LISTING 6: ATDISK.C

```
/*
 * ATDISK -- PC Tech Journal AT Hard Disk Performance Test
 * Version 1.00
 * Last modified 05/23/86
 * Copyright (c) 1986, PC Tech Journal
 * Program by: Paul Pierce, Ted Forgeron, Steven Armbrust
 * Uses BIOS INT 13H to measure low-level performance.
 * Uses DOS  INT 21H to measure high-level performance.
 */
#include "dos.h"
/* System clock tick rate in ticks/second */
#define TICK_RATE 18.20648
/* Number of trials for each test */
#define TRIALS 1000

struct diskparam {
        int     sectors;
        int     heads;
        int     cylinders;
        int     drives;
} diskparam[10];

char buffer[50 * 512];

/* Number of files for DOS file I/O test */
#define FILES   10
/* Size of each file for DOS file I/O test */
#define FILESIZE 20000
/* Size of each read/write request for DOS file I/O test */
#define FILEBUF 2000

char filename[] = "C:ATDISK.~~0";

unsigned long tick();
unsigned rand();

double drand()
{

        return (double)rand() / 32767;
}

#define READ_SECTORS    2
#define GET_PARAM       8
#define SEEK            12

main(argc, argv)
int argc;
char *argv[];
{
        register i, j, k;
        unsigned long start;
        unsigned long total;
unsigned trials;
int      f;

trials = TRIALS;
/*
 * Get the disk drive parameters for each hard disk
 * in the system.
 */
disk_io(GET_PARAM, 0x80, &diskparam[0], 0, 0, 0, 0);
for (i = 1; i < diskparam[0].drives; i++) {
        disk_io(GET_PARAM, 0x80 + i, &diskparam[i],
                0, 0, 0, 0);
}
/*
 * Display the disk characteristics.
 */
printf("\nATDISK -- PC Tech Journal AT Hard Disk ");
printf("Performance Test");
printf("\nVersion 1.00, Copyright (c) 1986 PC Tech ");
printf("Journal");
printf("\n\nHARD DISK CHARACTERISTICS\n");
printf("Drive     Sectors  Heads  Cylinders  Total ");
printf("space\n");
for (i = 0; i < diskparam[0].drives; i++) {
        total = diskparam[i].sectors * diskparam[i].heads;
        total *= diskparam[i].cylinders;
        total *= 512;
        printf("%3d       %5d   %5d     %5d     %10lu\n",
                        i,
                        diskparam[i].sectors,
                        diskparam[i].heads,
                        diskparam[i].cylinders,
                        total);
}
printf("\nBIOS TEST OF DISK HARDWARE PERFORMANCE\n");
/*
 * Seek back and forth between tracks 0 and 1.
 */
for (i = 0; i < diskparam[0].drives; i++) {
        disk_io(SEEK, 0x80 + i, buffer, 1, 0, 0, 0);
        start = tick();
        for (j = 0; j < trials/2; j++) {
                disk_io(SEEK, 0x80 + i, buffer, 1, 0, 0,
                        1);
                disk_io(SEEK, 0x80 + i, buffer, 1, 0, 0,
                        0);
        }
        total = tick() - start;
        printf("Drive %d track-track seek time: %4.1f ms",
                i, (double)total / ((double)trials) *
                  1000.0 / TICK_RATE);
        printf("\n");
}
/*
 * Seek to random tracks.
 */
for (i = 0; i < diskparam[0].drives; i++) {
        disk_io(SEEK, 0x80 + i, buffer, 1, 0, 0, 0);
        start = tick();
        for (j = 0; j < trials; j++) {
                disk_io(SEEK, 0x80 + i, buffer, 1, 0, 0,
                        (short)(drand()*
                        (diskparam[i].cylinders - 1)));
        }
        total = tick() - start:
```

The LAN Rush is On!



Get a QuickStart... Networking with TiaraLink™

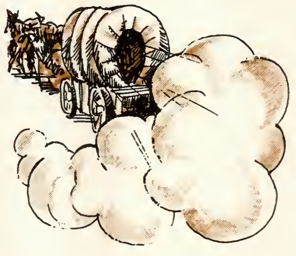
In today's business race, swift communication means the difference between being a success or an also-ran. Local area networks are the key to sharing important business data and resources. But, all too often potential LAN customers are stopped in their tracks by lack of experience.

QuickStart, a new feature, allows users to easily set up the powerful TiaraLink™ LAN — even with no programming or networking experience. This means fast and easy installations. Simple demonstrations. Less sales support. And more satisfied customers.

Now TiaraLink supports DOS 3.1 and a complete NetBIOS. So network application software written to these standards runs on TiaraLink. Already thousands of our networks are installed on IBM* PC, XT, AT and compatibles. TiaraLink is a **proven** LAN, and more... TiaraLink is the price/performance leader.

Tiara provides more than a high performance network — we offer you an outstanding dealer program: training, support, advertising, and some of the best margins in the industry.

The LAN Rush is on! So, don't be left at the starting line. Investigate the LAN that offers high performance at the best margins: TiaraLink. If you are interested in joining the Tiara dealership team, call us at **1-800-423-1268**. In California call **1-800-325-6223**. Or write us for more facts on TiaraLink.



Our Prices Leave Everybody Else in the Dust!

TIARA
COMPUTER SYSTEMS, INC.

2685 Marine Way • Mountain View, CA 94043 • (415) 965-1700 • TLX 4996251 • FAX (415) 965-2677

Trademark/Owners: TiaraLink/Tiara Computer Systems, Inc.; IBM/International Business Machines, Corp.

CIRCLE NO. 246 ON READER SERVICE CARD

```
ntf("Drive %d average seek time: %4.1f ms\n",
        i, (double)total / ((double)trials) *
           1000.0 / TICK_RATE);

 Measure transfer rate by repeatedly reading sectors
 */
for (i = 0; i < diskparam[0].drives; i++) {
        disk_io(READ_SECTORS, 0x80 + i, buffer, 1, 0, 0,
                0);
        start = tick();
        for (j = 0; j < trials; j++) {
                disk_io(READ_SECTORS, 0x80 + i, buffer,
                        diskparam[i].sectors, 0, 0, 0);
        }
        total = tick() - start;
        printf("Drive %d effective transfer rate: ", i);
        printf("%6.1f Kbytes/second\n",
                (((double)diskparam[i].sectors)*512.0) *
                (double)trials / ((double)total) /
                1024 * TICK_RATE);
}
/*
 * This section indicates overall system disk performance
 * by using DOS calls to manipulate files.  It is only
 * executed if the -d command line switch is used since
 * the test writes to disk.
 */
if (argc == 1) {
        printf("\nType \"atdisk -d\" to run the DOS ");
        printf("TEST OF DISK SYSTEM PERFORMANCE.\n");
        printf("This test creates, writes, reads and ");
        printf("deletes ten files named \n");
        printf("atdisk.~~0 through atdisk.~~9.\n");
}
if (argc > 1 && argv[1][0] == '-' && argv[1][1] == 'd'){
        printf("\nDOS TEST OF DISK SYSTEM PERFORMANCE\n");
        printf("This test creates, writes, reads and ");
        printf("deletes ten 20,000 byte files.\n");
        printf("Best results are achieved with blank, ");
        printf("freshly-formatted disks.\n");
                for (k = 0; k < diskparam[0].drives; k++) {
                        filename[0] = 'C' + k;
                        start = tick();
                        /*
                         * First create each file, write to it,
                         * and close it.
                         */
                        for (i = 0; i < FILES; i++) {
                                filename[11] = '0' + i;
                                f = dos_create(filename, 0);
                                for (j = 0; j < FILESIZE/FILEBUF;
                                     j++) {
                                        dos_write(f, buffer,
                                                FILEBUF);
                                }
                                dos_close(f);
                        }
                        /*
                         * Now reopen each file, read it, and
                         * then delete it.
                         */
                        for (i = 0; i < FILES; i++) {
                                filename[11] = '0' + i;
                                f = dos_open(filename, 0);
                                for (j = 0; j < FILESIZE/FILEBUF;
                                     j++) {
                                        dos_read(f, buffer,
                                                FILEBUF);
                                }
                                dos_close(f);
                                dos_delete(filename);
                        }
                        total = tick() - start;
                        printf("Drive %c:",'C' + k);
                        printf("%6.1f seconds\n",
                                ((double)total) / TICK_RATE);
                }
        }
}
/*
 * DOS file calls
 *
 * Each of these routines sets up the regs structure as
 * specified in the DOS technical reference and then
 * calls DOS using the intdos() function.
 */
/*
 * DOS call 3C - create file
 */
int
dos_create(file, attribute)
        char    *file;
        int     attribute;
{
        struct REGS regs;

        regs.x.dx =(int)file;
        regs.h.ah = 0x3C;
        regs.x.cx = attribute;
        intdos(&regs, &regs);
        if (regs.x.cflag) {
                printf("DOS error %d in create\n",
                        regs.x.ax);
                exit(1);
        }
        return regs.x.ax;
}
/*
 * DOS call 3D - open file
 */
int
dos_open(file, mode)
        char    *file;
        int     mode;
{
        struct REGS regs;

        regs.x.dx = (int)file;
        regs.h.ah = 0x3D;
        regs.h.al = mode;
        intdos(&regs, &regs);
        if (regs.x.cflag) {
                printf("DOS error %d in open\n",
                        regs.x.ax);
                exit(1);
        }
        return regs.x.ax;
}
/*
 * DOS call 3E - close file
 */
dos_close(handle)
        int     handle;
{
        struct REGS regs;

        regs.x.bx = handle;
        regs.h.ah = 0x3E;
        intdos(&regs, &regs);
        if (regs.x.cflag) {
                printf("DOS error %d in close\n",
                        regs.x.ax);
                exit(1);
        }
}
/*
 * DOS call 3F - read file
 */
int
dos_read(handle, buffer, count)
        int     handle;
        char    *buffer;
        int     count;
{
        struct REGS regs;

        regs.x.bx = handle;
        regs.x.cx = count;
        regs.x.dx = (int)buffer;
        regs.h.ah = 0x3F;
        intdos(&regs, &regs);
```

```
    (regs.x.cflag) {
            printf("DOS error %d in read\n",
                    regs.x.ax);
            exit(1);
    }
    return regs.x.ax;
}
/*
 * DOS call 40 - write file
 */
int
dos_write(handle, buffer, count)
        int     handle;
        char    *buffer;
        int     count;
{
        struct REGS regs;

        regs.x.bx = handle;
        regs.x.cx = count;
        regs.x.dx = (int)buffer;
        regs.h.ah = 0x40;
        intdos(&regs, &regs);
        if (regs.x.cflag) {
                printf("DOS error %d in write\n",
                        regs.x.ax);
                exit(1);
        }
        return regs.x.ax;
}
/*
 * DOS call 41 - delete file
 */
dos_delete(file)
        char    *file;
{
        struct REGS regs;
        regs.x.dx = (int)file;
        regs.h.ah = 0x41;
        intdos(&regs, &regs);
        if (regs.x.cflag) {
                printf("DOS error %d in delete\n",
                        regs.x.ax);
                exit(1);
        }
}
```

LISTING 7: PDISKIO.ASM

```
        NAME    PDISKIO
        PAGE    60,132
;*****************************************************************;
;                                                                 ;
;       PHYSICAL DISK I/O                                         ;
;                                                                 ;
;*****************************************************************;

_TEXT   SEGMENT  BYTE PUBLIC 'CODE'
_TEXT   ENDS
CONST   SEGMENT  WORD PUBLIC 'CONST'
CONST   ENDS
_BSS    SEGMENT  WORD PUBLIC 'BSS'
_BSS    ENDS
_DATA   SEGMENT  WORD PUBLIC 'DATA'
_DATA   ENDS
DGROUP  GROUP   CONST,  _BSS,   _DATA
        ASSUME  CS: _TEXT, DS: DGROUP, SS: DGROUP, ES: DGROUP

_TEXT           SEGMENT

;*****************************************************************;
;                                                                 ;
;       char                                                      ;
;       disk_io(cmd, drive, buf, count, sector, head, cylinder)   ;
;               4     6     8    10     12      14    16          ;
;*****************************************************************;

                PUBLIC  _DISK_IO
_DISK_IO        PROC    NEAR

        PUSH    BP                              ; SAVE FRAME
        MOV     BP, SP                          ; MAKE NEW FRAME
        PUSH    ES                              ;
        PUSH    SI                              ;
        PUSH    DI                              ;
        MOV     AH, [BP+4]                      ; PICK UP COMMAND
        MOV     DL, [BP+6]                      ; PICK UP DRIVE
        MOV     BX, [BP+8]                      ; PICK UP BUFFER
        PUSH    DS                              ;
        POP     ES                              ;
        MOV     AL, [BP+10]                     ; PICK UP COUNT
        MOV     DH, [BP+14]                     ; PICK UP HEAD
        MOV     CH, [BP+17]                     ; HIGH CYL BITS
        SHR     CX, 1                           ; SHF 2 MSB TO CL
        SHR     CX, 1                           ;
        AND     CL, 0C0H                        ; MASK THEM
        ADD     CL, [BP+12]                     ; ADD IN SECTOR
        INC     CL                              ; PLUS ONE
        MOV     CH, [BP+16]                     ; REST CYL IN CH
        INT     13H                             ; CALL BIOS
        JC      SAVERR                          ; JUMP IF ERROR
        MOV     AH, 0                           ; CLEAR RESULT
        CMP     BYTE PTR [BP+4], 8              ; DRIVE PARAM CMD
        JNE     SAVERR                          ; JUMP IF NOT
        MOV     BX, [BP+8]                      ; RESULTS IN BUF
        MOV     [BX], CL                        ; SECTORS
        AND     BYTE PTR [BX], 03FH             ;
        MOV     BYTE PTR [BX+1], 0              ;
        INC     DH                              ;
        MOV     [BX+2], DH                      ; HEADS
        MOV     BYTE PTR [BX+3], 0              ;
        MOV     [BX+4], CH                      ; CYLINDERS
        SHL     CX, 1                           ;  FIX HIGH BYTE
        SHL     CX, 1                           ;
        AND     CH, 003H                        ;
        MOV     [BX+5], CH                      ;
        MOV     [BX+6], DL                      ; DRIVES
        MOV     BYTE PTR [BX+7], 0              ;
SAVERR: MOV     AL, AH                          ; RESULT IN AX
        MOV     AH, 0                           ;
        POP     DI                              ; RESTORE REGS
        POP     SI                              ;
        POP     ES                              ;
        POP     BP                              ; RESTORE FRAME
        RET                                     ; RETURN

_DISK_IO        ENDP

;*****************************************************************;
;                                                                 ;
;       Get the current BIOS clock tick count                     ;
;       long ticks();                                             ;
;                                                                 ;
;*****************************************************************;

                PUBLIC  _TICK
_TICK           PROC    NEAR

        PUSH    BP                              ; SAVE FRAME
        MOV     BP, SP                          ; MAKE NEW FRAME
        PUSH    ES                              ;
        MOV     AX, 0                           ; CLEAR ES TO
        MOV     ES, AX                          ; GET BIOS DATA
        CLI                                     ; DISABLE INTS
        MOV     AX, ES:WORD PTR [046CH]         ; LOW TICK COUNT
        MOV     DX, ES:WORD PTR [046EH]         ; HIGH TICK COUNT
        STI                                     ; ENABLE INTS
        POP     ES                              ; RESTORE REGS
        POP     BP                              ; RESTORE FRAME
        RET                                     ; RETURN

_TICK           ENDP

;*****************************************************************;

_TEXT   ENDS

        END
```

THE PROGRAMMER'S SH

helps save time, money and cut frustrations. Compare, evaluate, and find proc

RECENT DISCOVERY

Quick Basic v2.0 - New user interface. Editor, source level error detection, built-in debugger, in-memory compilation, build and link user libraries. BASICA, GW-BASIC compatible. BLOAD, BSAVE PC $ 79

AI-Expert System Dev't

Arity System - incorporate w/C. PC $295
Experteach - Powerful, no limit on memory size. Samples PC $399
EXSYS - All RAM, Probability. PC $339
Insight 2 - dB2, language. MS $879
Others: APES ($359), Advisor ($949), ES Construction ($100), ESP ($845),

AI-Lisp

BYSO - Common, MacLISP compatible. 250+ functions, fast PC $150
GC LISP Interpreter - "Common", rich, Interactive Tutorial Call
Microsoft MuLisp 85 $ 250 $199
TLC LISP - classes, **compiler.** MS $225
TransLISP - Good for learning MS $ 75
WALTZLISP CPM-80 MS $149
Others: IQ LISP ($155), UNIX LISP ($59), IQC LISP ($269).

AI-Prolog

ARITY Standard - full, 4 Meg
Interpreter - debug, C, ASM PC $ 350
COMPILER/Interpreter-EXE PC $ 795
With Exp Sys, Screen - KIT PC $1250
MacProlog MAC $ 295
MicroProlog - enhanced MS $ 229
Professional MicroProlog MS $ 359
Prolog-86 - Learn Fast MS $ 95
Prolog-86 Plus - Develop MS $ 250
Others: Prolog-I ($365), Prolog-2 ($1795)
TURBO PROLOG by Borland PC $ 79

National Accounts Center

Special service is provided by a separate team for organizations with over 500 employees.

Purchasing agents, evaluators, managers, and programmers all appreciate the extra information, attention to shipping and invoicing, and help finding and evaluating products. Call 800-446-1185.

Our Services:

- **Programmer's Referral List**
- **Compare Products**
- **Help find a Publisher**
- **Evaluation Literature FREE**
- **BBS - 7 PM to 7 AM 617-826-4086**
- **Dealers Inquire**
- **Newsletter**
- **Rush Order**
- **Over 700 products**
- ***National Accounts Center***

C Language-Compilers

Aztec C86 - Commercial PC $499
Aztec C65 - Personal Apple II $199
C86 by CI - 8087, reliable MS $299
Lattice C - from Lifeboat MS $289
Lattice C - from Lattice MS $339
Mark Williams - w/debugger MS $399
Microsoft C 3.0 - new MS $259
Q/C 88 by Code Works - Compiler source, decent code, cross/native MS $125
Wizard C - Lattice C compatible, full sys. III, lint, fast. MS $389

C Language-Interpreters

C-terp by Gimpel - full K&R MS $249
INSTANT C - Source debug, Edit to Run-3 seconds MS $399
Interactive C by IMPACC Associates. Interpreter, editor, source, debug. PC $225
Introducing C - self paced tutorial PC $109
Run/C Professional - Run/C plus create add-in libraries, more. MS $199
Run/C - improved MS $109

C Libraries-Communications

Asynch by Blaise $149
Greenleaf - full, fast $139
Software Horizons - pack 3 $119

C Support-Systems

Basic C Library by C Source $1
C Sharp - well supported. Source, **realtime,** tasks, state system $600
C ToolSet - DIFF, xref, source $ 95
Lattice Text Utilities $105
PC LINT - Checker AMIGA $89 MS $119
The HAMMER by OES Systems $179
SECURITYLIB - add encrypt to MSC. C86 programs. Source $250 PC $125

C Libraries-General

Application Programmer's Toolkit MS $375
Blaise C Tools 1 ($109), C Tools 2 $ 89
C Essentials - 200 functions PC $ 85
C Food by Lattice-ask for source MS $109
C Power Windows by Entelekon $119
C Utilities by Essential - Comprehensive screen graphics, strings. Source. PC $139
C Worthy Library - Complete, machine independent, source MS $295
Entelekon C Function Library PC $119
Greenleaf Functions - portable, ASM $139
PforCe by Phoenix - objects PC $299
Polytron - for Lattice, ASM source $ 85

C Libraries-Files

FILES: C Index by Trio - full B+ Tree, vary length field, multi compiler
/File is object only $ 89
/Plus is full source $349
C to dBASE - with source MS $139
CBTREE - multiuser record locking, sequential, source, no royalties MS $ 99
CTree by Faircom - no royalties MS $339
dbVISTA - full indexing, plus optional record types, pointers, Network.
Object only - MS C, LAT, C86 $159
Source - Single user MS $429
Source - **Multiuser** MS $849
dBASE Tools for C PC $ 79
dbc Isam by Lattice MS $199

FEATURE

Multitasking, Windowing for C, Turbo Pascal, or dBASE or . . . in only 12K!
SYNERGY

The highly efficient design of Synergy by Matrix gives you the benefits of powerful graphics, windows, pull-down menus, dialog boxes, sophisticated text and icon management, math support, multitasking, and SPEED, all for an incredibly small **12K RAM** requirement.

The Synergy Runtime provides character and graphics support for menus, windows, dialog boxes, and more, so you can write programs that work in either mode, with very reasonable, low runtime fees.

Functions include: window management with capabilities like tile and overlap, variable size and placement, process management, to support multitasking and sub-process generation, menus, dialog, and icon management, graphics, text (including a variety of typefaces, fonts, and sizes), and console management.

The Synergy Development Toolkit is a collection of sophisticated tools designed for software developers writing new applications using Synergy, or modifying existing applications to take full advantage of Synergy. Tools include: graphics resource editor for creating and modifying icons and text fonts, graphics resource compiler to construct and manage resource data files, font compiler and manager, debugging tools, sample library, and more.

Supports IBM or Microsoft Macro Assemblers, Turbo, IBM, and Microsoft Pascal, IBM and Microsoft BASIC, Lattice and Microsoft C, and dBASE II and III. CGA, EGA, and Hercules monochrome graphics support.

PCDOS $395

FEATURE

Multi-User Data Mangement
C-INDEX B+TREE LIBRARY

— "Had I known about your product when I was designing and writing dBASE III, I would have certainly used it." — Wayne Ratliff, dBASE author

C-Index can enhance your product development by providing powerful, time tested data management, straight out of the box.

The C-Index Library delivers high performance B+tree indexing with efficient variable length record storage. You get full transportable source code, pre-compiled object libraries, and there are no royalty charges. Additional features include record locking, random and sequential data access, automatic multi-key maintenance, and virtual memory buffering. Single or multiuser, LAN and Xenix compatible.

Highly portable: IBM PC/AT, MacIntosh, Cray, VAX, more.

C-Index/Plus (MSDOS, MAC): $349
C-Index/File (MSDOS) object only: $ 89

FEATURE

Oops Protection Plus Complete Disk Control with
Disk Toolkit

UNERASE is valuable, but when your file is damaged, it does not work. You need Disk Toolkit's:

- Full-screen FAT table editor
- ASCII Search (or HEX)
- Disk verification to find bad sectors and examine data
- File Allocation Table and full directory access and control
- Ability to copy any sector, track, cluster, file, or area of memory to . . . anywhere — your file on another disk
- Unerase — it can help. Operate in beginner's (keystroke) or advanced mode
- Work on any size hard disk

Disk Toolkit is complete, practical, small (25K), and guaranteed.

PCDOS $75

We support MSDOS (not just compatibles), PCDOS, Xenix-86, CPM-80, Macintosh, Atari ST, and Amiga.

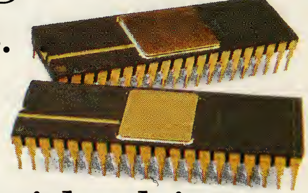
CIRCLE NO. 162 ON READER SERVICE CARD

coprocessor will be going fast.

Like the new Above Board PS/AT.

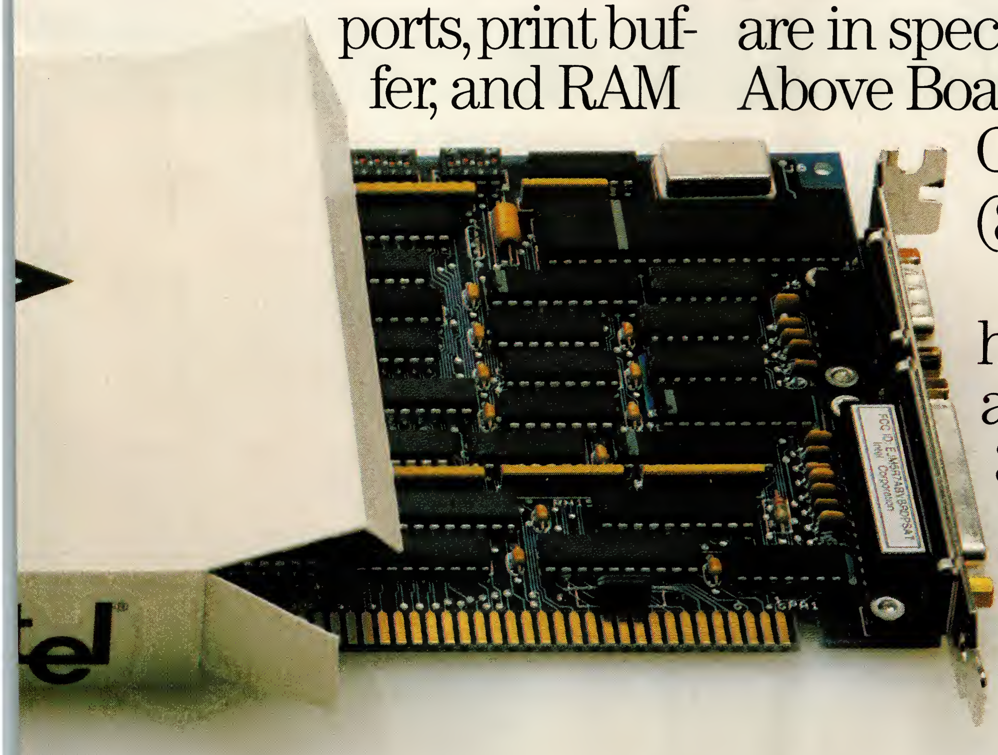
It gives you expanded memory—which meets the Lotus®/Intel/Microsoft® spec, conventional memory—up to 640K, and extended memory—capable of supporting protected mode operating systems.

Plus serial and parallel ports, print buffer, and RAM disk. All packaged into a single slot. And all guaranteed to run at 8 MHz.



Of course there are also Above Boards that provide this same memory—with and without ports—for your PC/AT, XT and compatibles.

Details on how to get your free coprocessor chip are in specially marked Above Board packages.

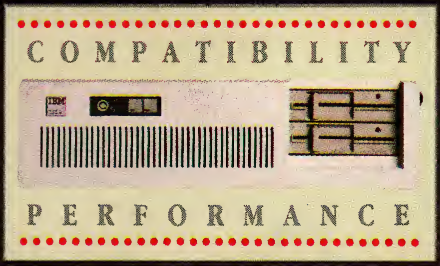


Or call us at (800) 538-3373.

But you'd better hurry. Because after September 30, this offer will be long gone.

intel®

CIRCLE NO. 216 ON READER SERVICE CARD



OUT FROM THE SHADOW OF IBM:

Compaq Deskpro 286

Operated as either a 6-megahertz or an 8-megahertz machine, the Deskpro 286 carries on Compaq's tradition of overall quality and IBM compatibilty.

STEVEN ARMBRUST and TED FORGERON

From the introduction of its first computer, the Compaq Portable, Compaq has gained a reputation for quality and compatibility that has enabled its computers to be accepted and used wherever IBM PCs are used. It therefore seems appropriate that a computer from Compaq should be the first one tested in *PC Tech Journal*'s series on AT-compatible machines. The Deskpro 286, like other Compaq machines, is a rugged, high-quality computer that can be an ideal alternative to the AT for users who need some of the extra features provided. A speed selection switch on the Deskpro 286 allows users to change from 6-MHz to 8-MHz on the same machine, a definite advantage over IBM. Other extras include an integrated tape backup system, dual-mode monitor and adapter that displays high-resolution text characters and produces graphics compatible with the IBM Color Graphics Adapter.

Compaq offers three models of the Deskpro 286, each distinguished by the amount of memory and the kinds of peripherals. The top-of-the-line model 3 was used for this review. This machine was equipped with 512KB of memory, a 30MB hard disk, 1.2MB diskette drive, 10MB tape backup system, asynchronous communications port, parallel printer port, and single-color monitor connected to Compaq's video display controller card. The features available with all three Deskpro 286 models are listed in the accompanying sidebar, "Deskpro 286 Vital Statistics."

Unlike some of the other AT-compatible computers on the market, the Deskpro 286 does not try to mimic exactly the exterior styling of the AT. Instead, it conforms to the style introduced with the original Compaq Portable and is almost a dead ringer for the 8086-based Deskpro. The system unit of the Deskpro 286 measures 19¾ inches by 16½ inches by 6¼ inches. Photo 1 compares its footprint to that of the AT. (The photo shows the top of the AT sys-

PHOTOGRAPH • MARC DAVID COHEN

PHOTO 1: *System Unit Footprint*

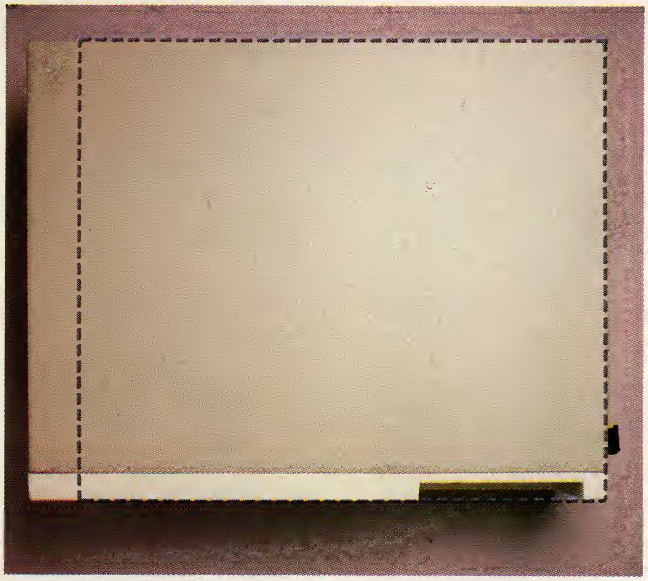


PHOTO 4: *Inside the System Unit*

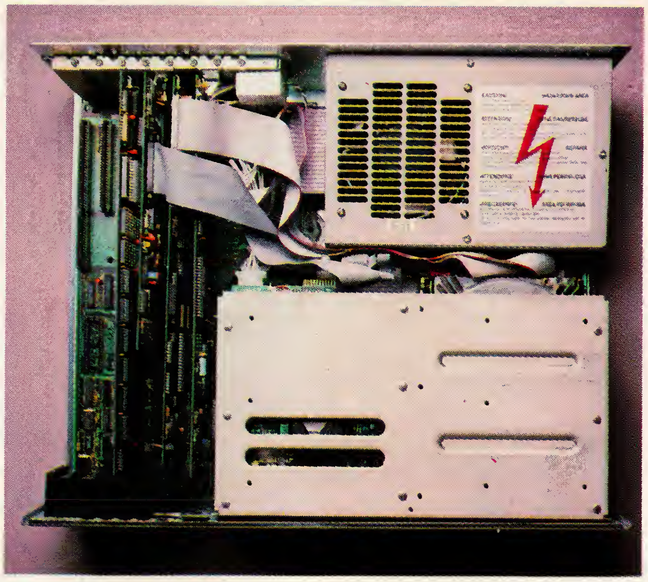


PHOTO 2: *Keyboard Comparison*

PHOTO 5: *Head-locking Device*

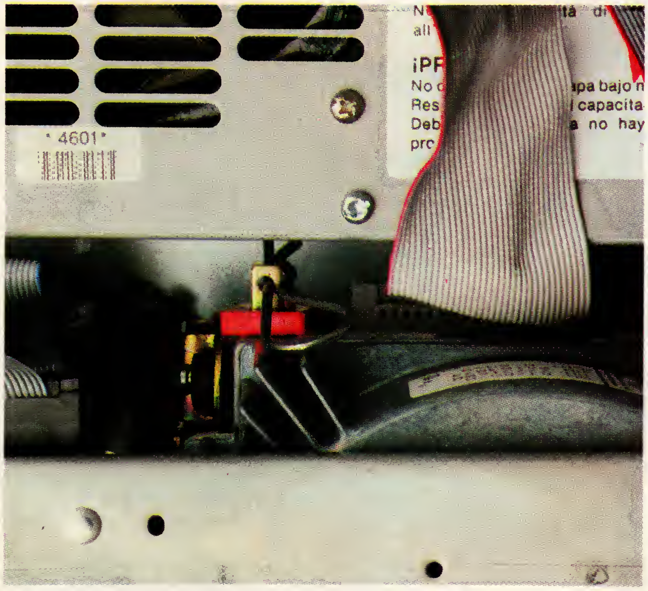


PHOTO 3: *Rear Panel*

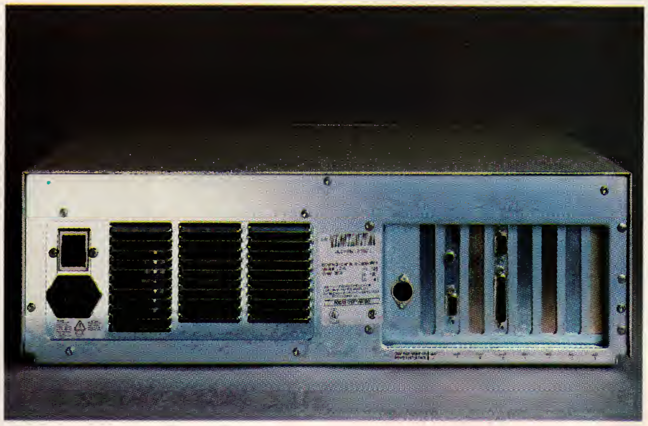


Photo 1: Compaq did not try to imitate the shape of the IBM AT. Its system unit is 19¾ by 16½ by 6¼ inches, indicated by dotted lines, compared to 21¼ by 16½ by 6¼ for the AT.

Photo 2: The Deskpro 286 keyboard matches the AT keyboard in most respects, but does not have the solid IBM feel. The status lights are conveniently placed on the keys.

Photo 3: The location of the power switch on the rear panel is inconvenient; the front panel would have been a better position. The 4½-inch fan provides plenty of air flow.

Photo 4: The math coprocessor socket is easy to reach, but a metal brace between the front and rear panels of the system unit blocks convenient access to slot number 1.

Photo 5: Before using the hard-disk drive for the first time, the cotter pin and plastic washer must be removed, and the exposed lever must be pushed into its unlocked position.

tem unit with the Deskpro outline superimposed as a dotted line.) The Deskpro 286 is a big, heavy machine, indicative of its sturdy construction. The heavy-duty hard disk adds its not insignificant bulk to the load. Moving this computer around can be a chore.

Only two exterior features, a keylock switch and an AT-style keyboard, differ from the 8086 version. Neither of these features is up to the standards set by the similar items on the AT. Instead of the slick chrome keylock available on the AT, the Deskpro 286 has a dull black mechanism stuck incongruously to the front panel of the system unit with a key of the bus-station-locker variety. Part of the mechanism holding the lock in place fell off the test machine while the system unit cover was being removed, hardly inspiring confidence in its ability to deter intruders. An indicator for the locked and unlocked positions is so faint that it can barely be seen, even in direct light.

The Deskpro 286 keyboard is likely to be the source of many complaints for a variety of reasons. Although it conforms for the most part to the layout of the standard AT keyboard (see photo 2) and it offers more tactile feedback than the keyboard on the original Compaq Portable, it still lacks the solid feel of IBM. The oversized keys (Enter and Shift) are particularly flimsy. The computer generates audible keyboard clicks to compensate for the keyboard's lack of tactile feedback. The Ctrl-Alt-minus and Ctrl-Alt-plus key combinations reduce and increase the loudness of the key clicks. (The Deskpro 286's BIOS keeps track of the loudness by storing a value between 0 and 127 at absolute RAM address 416H. The IBM BIOS reserves this byte as a scratch area.)

The keyboard's adjustable legs, which are the same kind that were used on earlier Compaq models, do not lock into position, so any abrupt or sideways motion can cause those legs to collapse. The Compaq 286's keyboard cord also suffers from clumsiness. It is approximately the same length as the one on the AT, but because the Compaq keyboard plugs into the front of the system unit instead of the back as on the AT, the extra cord length tends to get in the way. Perhaps Compaq has overreacted to the criticisms of the Compaq Portable's too-short cord.

The keyboard that arrived with the test machine had problems with sticking keys, particularly Alt and Enter. Whenever the Ctrl-Alt-Del combination was pressed, the system displayed a keyboard error message on the screen instead of rebooting. This keyboard was replaced with another, but after typing a single word into a word processor with the new keyboard, a never-ending stream of spaces spewed out onto the screen, obliterating text at 8 MHz. The Space bar lay plastered to the base of keyboard and only with great effort could it be pried up. This is not to say that all Deskpro keyboards are plagued with sticking keys, but a cautionary note would be to test all the keys before taking a new machine home.

The location of the power switch on the system unit might cause some frustration. Instead of following IBM's lead and placing the switch on the right side of the unit (or better yet, placing it in front for easy access), Compaq put the switch on the rear panel (see photo 3). This location certainly minimizes the risk of accidentally turning off the machine, but it forces the user to reach around to the back and fumble blindly for an invisible switch.

The Deskpro's monochrome display is an example of high-quality construction and good design. Used with Compaq's standard display adapter, which emulates IBM's Color Graphics Adapter and generates high-resolution text comparable to that generated by IBM's monochrome adapter, it displays crisp characters and graphics in an eye-pleasing shade of amber. An especially nice feature of the display is its viewing angle. Because of its almost pyramid shape, when the display sits on a horizontal surface, the screen tilts slightly upward. This permits comfortable viewing either when the entire machine is sitting on a typing table or when the display is sitting directly on a desktop.

Compaq's 1.2MB diskette drive has a noteworthy feature. When the drive is reading or writing a 1.2MB diskette, the drive light shines green; when the drive accesses a 360KB diskette, the light turns to red. With all the juggling between diskette formats that goes on these days, the different colors can be reassuring, especially when formatting.

Like the AT's 1.2MB diskette drive, the Deskpro 286's 1.2MB drive does not always write 360KB diskettes reliably. Moreover, during testing, the drive sometimes had trouble *reading* 360KB diskettes. For example, the drive would not read two of the Lotus Symphony master diskettes. However, when a PC with 360KB drives was used to make copies of those diskettes, the Deskpro was able to read the copies.

Standard on the Deskpro 286 is a parallel printer port and an asynchronous communications adapter (serial port), which are both on the diskette adapter card. The connectors are identical to those on the AT; the parallel port has a 25-pin, D-shell, female connector, and the serial port has the smaller AT-style 9-pin, D-shell, male connector. Jumpers on the diskette adapter card allow switching or disabling the ports.

DESKPRO 286 VITAL STATISTICS

Model 1: $3,999
256KB memory
Parallel printer interface
Serial interface
Dual-mode (monochrome and CGA compatible) display adapter
RGB and composite interface
Dual-mode monochrome monitor
Realtime clock
1.2MB or 360KB diskette drive

Model 2: $4,999
All features of model 1 plus:
512KB memory
30MB hard disk

Model 3: $5,699
All features of model 1 plus:
512KB memory
30MB hard disk
10MB tape backup

Memory capacity on system board
None

Display adapters
Dual-mode (IBM monochrome and CGA compatible)

Expansion slots
16-bit: 4
8-bit: 2

Available slots
Model 1
16-bit: 3
8-bit: 2
Models 2 and 3
16-bit: 2
8-bit: 2

Extras available
10MB tape backup
Processor speed selection switch (6 MHz/8 MHz)
Dual-mode display adapter and monitor (monochrome/CGA)

The *Operations Guide* contains no information about these jumpers, but a telephone call to Compaq revealed that the first three jumpers are for enabling or disabling COM1, COM2, and PRN, respectively. The fourth one is removed in order to change the primary address for the floppy-diskette controller from 3F0H to 370H so two more floppy-disk drives can be added to the machine. This information is in the *Maintenance and Service Guide*, which is not supplied with the machine but is available as an option.

The computer tested for this article arrived with Compaq's own hard-disk controller card and a 30MB hard disk. This is recommended because it gives users a high-quality, high-performance disk that is guaranteed to work in their machines. A user who wishes to buy a third-party hard disk should realize that, unlike the AT, the base model Deskpro 286 does not contain a hard-disk controller. Several controllers are available, but Compaq has not tested any of them with the Deskpro 286. The user should test drive a third-party controller before purchasing one.

EXTRAS, EXTRAS!

In addition to all the standard AT features, the Deskpro 286 includes several extras not found in any IBM models. The most notable of these is the tape backup system.

To address the problem of backing up data from a 30MB hard disk, the Deskpro 286 model 3 includes a tape backup unit capable of reading and writing the small DC-1000 tape cartridges from 3M. Although this unit seems to work reliably, it has two problems that suggest it will not become a favorite among users. It is relatively slow, and it cannot back up the entire hard disk on a single tape.

Formatting a tape is normally a time-consuming operation, and the Deskpro 286's tape-formatting is no exception. Approximately 20 minutes are required to format each 10MB tape, and another 20 minutes to fill that tape with backup data. That is one hour to format enough tapes to back up the complete 30MB hard disk, and another hour to copy the data.

Lack of speed would not be a crucial deficiency for the tape drive if the user could invoke the tape backup utility and then walk away while the unit continued to capture the data on the disk. However, each cartridge stores only 10MB of data, so backing up a 30MB hard disk requires at least three tape changes during the process.

Another special feature that all Compaq models have is the dual-mode monitor and display adapter. This monitor/adapter combination allows text to be displayed with the same resolution as IBM's monochrome adapter/monochrome monitor combination. Other characteristics of the monochrome adapter, such as underlining and reverse video, also are supported. The dual-mode monitor/adapter is compatible with the IBM CGA, allowing graphics to be displayed on the screen and a third-party color monitor to be connected.

The Compaq adapter is really just a CGA with the extra capability to support high-resolution characters. Like the CGA, it contains 16KB of video memory starting at address B8000H, and it supports video modes 0 through 6 for 40-column text (modes 0-1), 80-column text (modes 2-3), and graphics (modes 4-6). Even though it can generate high-resolution characters, it does not support mode 7, used by IBM's monochrome adapter.

Compaq allows users to switch the processor speed between 6 MHz and 8 MHz by pressing Ctrl-Alt-\. This accommodates those programs that simply will not work at higher speeds.

The key to Compaq's ability to display high-resolution characters is in the character generator on the display adapter. Whereas in graphics mode, characters are formed by copying dot patterns that are stored in the system ROM BIOS, in the text modes the character generator is used. In the 80-column text modes, the generator can produce two different kinds of characters, and the dual-mode monitor is capable of displaying either kind.

The two kinds of characters correspond to the high-scan and low-scan modes of the monitor. In low-scan mode, the video signal sent by the display adapter uses a horizontal scanning frequency of 15.7 KHz and a vertical scanning frequency of 60 Hz, noninterlaced. These frequencies, which are the same ones generated by the CGA, are compatible with color monitors. The display adapter sends characters that use an 8-by-8 dot pattern, just like those generated by the CGA. In addition, the attribute byte associated with each character is the standard color attribute byte (indicating the foreground and background color, the intensity, and whether the character is blinking). Low-scan mode is the only one used for the 40-column text and graphics modes.

In high-scan mode, the video signal sent by the display adapter uses a horizontal scan frequency of 18.5 KHz and a vertical scan frequency of 50 Hz, noninterlaced. The display adapter sends characters that use a 9-by-14 dot pattern, just like those generated by the IBM monochrome display adapter. In addition, the attribute byte associated with each character is the same as that used with the IBM monochrome display adapter (allowing normal, reverse video, invisible, underlined, blinking, and high-intensity characters).

By default, whenever the Compaq is in the 80-column video modes, it displays the characters in high-scan mode. Therefore, even though programs think they are dealing with an ordinary CGA in text mode, the characters that appear on the screen are the higher-resolution variety that is normally associated with the IBM monochrome adapter.

A user can switch between high-scan and low-scan modes by pressing Ctrl-Alt-> (for high-scan) and Ctrl-Alt-< (for low-scan). This can be done while running any program, because an individual program will notice no difference between the two modes. Switching to low-scan mode allows text to be displayed on a third-party color monitor.

Compaq allows users to switch the processor speed between 6 and 8 MHz by pressing Ctrl-Alt-\. This accommodates those programs that just will not work at higher speeds (although with the AT now available in an 8-MHz version, fewer of these programs will be apparent). Once the key combination is pressed, the machine announces the mode by beeping (once for 6 MHz, twice for 8 MHz). The same beeping announcements occur when booting the computer. Although this is a clever way to indicate the processor speed, the double-beep during boot-up is very similar to the BIOS error beeps that occur when the PC experiences a parity error. A more distinctive sound would prevent veteran PC users from experiencing a sudden rush of anguish the first few times they boot their Deskpros.

The processor speed also can be set using the DOS MODE command, which offers an additional (and for-

TABLE 1: *Compatibility and Performance Tests*

	IBM AT with 30MB disk[a] (8-MHz model)	COMPAQ DESKPRO 286 with 30MB disk Fast[b]	High[c]	COMPAQ PORTABLE II with 10MB disk Fast	High
ATBIOS					
ROM BIOS date	11/15/85	11/20/85	11/20/85	11/20/85	11/20/85
ATPERF					
Average RAM instruction fetch (μs)	0.403 (100)[d]	0.404 (100)	0.404 (100)	0.404 (100)	0.403 (100)
Average RAM read time (μs)					
BYTE	0.401 (100)	0.403 (99)	0.401 (100)	0.403 (99)	0.401 (100)
WORD	0.401 (100)	0.403 (99)	0.401 (100)	0.403 (99)	0.401 (100)
Average RAM write time (μs)					
BYTE	0.401 (100)	0.404 (99)	0.402 (100)	0.404 (99)	0.402 (100)
WORD	0.401 (100)	0.403 (99)	0.401 (100)	0.403 (99)	0.402 (100)
Average ROM read time (μs)					
BYTE	0.401 (100)	0.403 (100)	0.401 (100)	0.403 (100)	0.401 (100)
WORD	0.401 (100)	0.403 (100)	0.401 (100)	0.403 (100)	0.401 (100)
Average video write time (μs) (CGA only)					
BYTE	1.208 (100)	1.111 (109)	1.013 (119)	1.111 (109)	1.013 (119)
WORD	2.415 (100)	2.217 (109)	2.025 (119)	2.217 (109)	2.025 (119)
Average EMM read time (μs)					
BYTE	0.402 (100)	0.540 (74)	0.403 (100)	0.540 (74)	0.402 (100)
WORD	0.402 (100)	0.540 (74)	0.402 (100)	0.540 (74)	0.402 (100)
Average EMM write time (μs)					
BYTE	0.402 (100)	0.540 (74)	0.402 (100)	0.539 (74)	0.402 (100)
WORD	0.402 (100)	0.540 (74)	0.403 (100)	0.540 (74)	0.402 (100)
CPU clock rate (MHz)	8.0 (100)	7.9 (99)	8.0 (100)	7.9 (99)	8.0 (100)
Math coprocessor clock rate (MHz)	5.3 (100)	4.2 (78)	5.3 (100)	4.2 (78)	5.3 (100)
Refresh overhead (%)	7.1	6	7	6	7
RAM read wait states	1	1	1	1	1
RAM write wait states	1	1	1	1	1
ROM read wait states	1	1	1	1	1
Video write wait states (CGA)	8	7	6	7	6
EMM read wait states	1	2	1	2	1
EMM write wait states	1	2	1	2	1
ATFLOAT					
Performance as percentage relative to AT	100	80	100	80	100
ATDISK					
Sectors/track	17	17	17	17	17
Heads	5	5	5	4	4
Cylinders	731	695	695	304	304
Total space (million bytes)	31.81	30.25	30.25	10.58	10.58
Track-track seek time (ms)	9.9	6.4	6.4	16.9	16.8
Average seek time (ms)	37.0	40.2	40.5	71.6	71.6
Effective transfer rate (KB/sec)	170.1	256.2	256.2	255.0	255.0
DOS file I/O (sec)	7.3	8.1	9.1	10.0	9.3
Interleave	3	2	2	2	2

[a] *The figures for the IBM AT are the average results from several machines, whereas the results from the Compaq machines are taken only from the review sample model.*
[b] *Fast is the default speed.*
[c] *High is obtained by the MODE SPEED command.*
[d] *Figures shown in parentheses represent the relative performance expressed as a percentage compared to* PC Tech Journal's *baseline machine, the 8-MHz, 30MB AT.*

The 8-MHz Deskpro 286 has an 8-MHz processor and a 6-MHz bus in default mode. Compaq memory has special circuitry that allows it to be accessed with only one wait state. Third-party memory boards are accessed with two wait states. The DOS command MODE SPEED = HIGH, enables the Deskpro to operate as a full 8-MHz machine.

THE PORTABLE II: MEETING FAMILY EXPECTATIONS

The smallest member of the Compaq family, the new Portable II, compares favorably with its larger and heftier siblings. With a few exceptions, its compatibility and performance figures match those of the Deskpro 286.

Although the Portable II contains almost all the features of Compaq's earlier Portable 286, this redesigned model occupies almost one-third less space. By using one-third-height disk drives, providing only two extra expansion slots (one 8-bit and one 16-bit slot) and locating the function keys on a single horizontal row above the numeric keys, Compaq was able to squeeze an entire AT-compatible computer into a box that measures 13 inches deep, 17½ inches across, and 7¼ inches high. The review computer was a model 3 and contained a single 360KB diskette drive and a shock-mounted 10MB hard disk. It weighed in at a mere 23.6 pounds, 17 percent lighter than a comparably-equipped Portable 286. At this size and weight, carrying a computer through a busy airport and trying to store it under an airline seat at long last becomes a manageable proposition.

Compaq compromised little to achieve the smaller size. The screen is a 9-inch dual-mode monitor that is even a little larger than the ones that have graced previous Compaq portables. The diskette drive, a one-third-height model, uses ordinary 360KB 5¼-inch diskettes. Even the keyboard is an AT-style model, with the only difference being the position of the 10 function keys. Compaq chose not to provide a 1.2MB diskette drive with the Portable II, which may be a wise move judging from the difficulties the Deskpro 286 experiences in reading and writing 360KB diskettes.

The Portable II offers the same extra features as the Deskpro 286, such as the dual-mode monitor and speed selection switch (but not the tape backup system). However, Compaq has made some design improvements to address the most common complaints made of its earlier models. The Portable II has an acceptably long keyboard cable, which slides out from a deep trench under the diskette drive. It is long enough for the user to set the computer on a desk and lean back with the keyboard in his lap.

The adjustable legs on the computer have been redesigned. The keyboard legs open by tilting back, instead of to the side, making the keyboard less likely to collapse due to sideways jarring. The tilt-up legs are gone from the main unit too, replaced by a wide brace that doubles as a door to the power-cord container. This new design makes the unit much more stable and allows for convenient storage of the power cord.

The plastic outer shell on the Portable II is a single piece, so users

merly undocumented) high-speed mode that is not available with the keyboard sequences.

MODE SPEED=FAST is the 8-MHz mode described above. This default mode sets the CPU to 8 MHz and the bus to 6 MHz. MODE SPEED=COMMON is the 6-MHz mode that sets the CPU and bus to 6 MHz. Setting the mode via DOS does not provide any audible indication as does the Ctrl-Alt-\ key combination. The Deskpro 286 processor also can be set to run at 8 MHz with an 8-MHz bus using MODE SPEED=HIGH. This is not the default mode—it cannot be accessed using Ctrl-Alt-\, and it is undocumented in all but the most recent Compaq manuals (scheduled to be released this summer). Nonetheless, it is available to enable the machine to be run at its fastest speed. The current mode of the machine can be determined using the command MODE SPEED. The response will be FAST, COMMON, or HIGH.

EASE OF INSTALLATION

One of the most difficult parts of installing hardware in the Deskpro 286 involves the screws. The cover of the system unit is fastened with three screws on the rear panel, among a total of eight similar-looking screws, so selecting the right ones can be a guessing game. Furthermore, most of the screws on the Deskpro 286 are the Torx T-15 type, which are great for automating Compaq's manufacturing process, but difficult to remove with common screwdrivers (and few ordinary people actually own Torx screwdrivers).

The Deskpro's system unit cover is a wraparound design similar to that of the PC and AT. It slides forward and then lifts straight up and off. Because the keyboard is attached at the front of the unit, it must be removed before taking off the cover. (Nonetheless, this is much more accessible than fumbling around the back of an IBM model to attach and detach the keyboard.)

Despite these few inconveniences, installing hardware in the Deskpro 286 is, for the most part, an easy process.

who are used to prying the side walls off the old portables can save their fingernails. When the computer is sitting so the handle points up, the shell lifts straight up and off. However, some energetic prying is required to maneuver the plastic protrusions around the power switch.

Like the Deskpro 286, the Portable II uses Torx screws. The plastic case has a different size (T-15) than the aluminum panels and slot covers (T-10); neither set has grooves for flat blade screwdrivers. Compaq does not include screwdrivers of either size with the Portable II.

Unlike the Deskpro, the Portable II's system board has room for 640KB of RAM. In addition, Compaq offers a special memory board, with room for up to 1.5MB of memory, that plugs directly into the back of the system board instead of requiring an expansion slot. This extra memory can be treated only as extended memory, with addresses beginning at 1MB. By allowing for more than 2MB of memory connected directly to the system board, the competition for expansion slots is minimized.

The Portable II has only four expansion slots, of which two are in use. In the model 3 tested, slot 1 contains the controller for the hard disk and diskette drive. This controller also contains an asynchronous communications port and a parallel printer port. Slot 2 contains the dual-mode video display adapter. Slot 3 is a 16-bit and slot 4 is an 8-bit slot. This lack of expansion slots could be a problem for users who have many options to install. The Portable II is no different from other portable computers in this area, however. Small size and adequate room for expansion cards seem to be mutually exclusive.

Slot 4 on the Portable II is set apart from slot 3 to allow room for an internal modem's speaker—a natural option on such a machine. However, the space between slot 3 and the neighboring slot 2 is insufficient to install a piggyback card. This limitation prevents Intel's 4MB Above Board from being installed in the Portable II. The video card in slot 2 cannot be moved to the other 8-bit slot (slot 4) to make more room, because the ribbon cable connecting the display to the adapter is too short to allow the adapter to reside in the farther slot. Only the 2MB version of Above Board (without the piggyback) can be installed. This narrow slot will probably cause the same problem when installing any 16-bit card with a piggyback.

Another problem with installing Above Board is that the board is too tall to fit properly unless the plastic support bracket is removed from the Portable II's board slot access cover. Whether Compaq or Intel is at fault here is open to question. Users who plan to use expanded memory should be aware of the difficulties of squeezing the Intel board into the Portable.

The same compatibility and performance tests were run on the Portable II as were run on the Deskpro 286, and for the most part, the same results were obtained. The Portable II was able to run all the sample software and, except for the difficulty in installing Above Board, it worked with the add-on hardware products.

Like the Deskpro 286, the Portable II failed the video adapter test in the IBM AT Advanced Diagnostics, it ran the 80287 math coprocessor at only 4.2 MHz in the default mode, instead of the expected 5.3 MHz, and it used two wait states instead of one when accessing expanded memory, again in the default mode (indicating a 6-MHz instead of an 8-MHz expansion bus). But also like the Deskpro 286, accessing the video memory on the color/graphics adapter was 60 percent faster than on the AT. Full 8-MHz capabilities (expansion bus and CPU) are available in the same way as on the Deskpro 286 by using the DOS MODE SPEED=HIGH command.

The Portable II did not measure up to the Deskpro 286 in hard-disk performance. As table 1 in the main article shows, the seek time of the Portable II's 10MB drive, both track-to-track and average, was double that of the Deskpro 286's 30MB unit. These figures are comparable to a lower performance XT-class machine. Compaq recently announced a 20MB hard disk for the Portable II, but it was not available in time for this review.

The Portable II offers the state of the art in portable computing without the unpleasantness of LCD displays or the compatibility problems of 3½-inch diskettes. Although the 10MB drive does not measure up to the standards expected for 80286 computers, the nondisk performance figures match those of the Deskpro 286, and the Portable II is as compatible as any Compaq model. Its hard-disk deficiencies and lack of expansion slots preclude the Portable II from being effective as a user's primary, high-performance computer. But as a business computer to take on the road, or for transporting between home and office, it is an ideal solution.

—Steven Armbrust and Ted Forgeron

Photo 4 shows the interior of the system unit. Adding an 80287 math coprocessor is simple (even though the Deskpro's *Guide to Operations* manual does not mention the 80287 socket), because the socket is located under board slot 1 where it is quite easy to reach, rather than near the power supply as on the AT. One minor annoyance is a metal brace that fastens between the front and rear panels of the system unit and is positioned directly over slot 1; the brace must be taken out before any board can be installed in or removed from that particular slot.

The peripherals (diskette drive, tape drive, and hard disk) are surrounded by a sheet-metal enclosure, preventing any sensitive components or mechanical pieces from being jostled or damaged while working in other areas of the unit. In fact, the metal enclosure is solid enough to hold test equipment or additional drives that need to be tested before installation.

Because of the kind of hard-disk drive that Compaq uses (a rugged-looking piece of Control Data equipment), a manual head-unlocking step must be performed before using the computer initially. As shipped from the factory, the drive's locking lever is pulled out in its locked position and held there with a plastic washer and cotter pin arrangement (see photo 5). Before applying power to the computer, the user must remove the cotter pin and plastic, push the lever into its unlocked position, and plug the unit into the power supply (which could not be done before because the locking mechanism blocked the connector). A sticker on the front of the drive warns users that they must follow this procedure. Of course, when preparing the unit for moving, this process must be reversed in order to lock the heads again.

The manual head-locking procedure perhaps is less convenient than using software to lock the heads, but a certain comfort is derived from knowing—really knowing—that the drive's heads are securely locked during a bumpy move.

Orchid's TurboEGA™

The world's *fastest* EGA

FREE MS WINDOWS™

Introducing the only Enhanced Graphics Adapter with PCturbo™ speed.

The experts agree: the EGA is the breathtaking new graphics standard, but the sophisticated software written for it places a big burden on the PC's processing speed. Beautiful graphics, crisp text, but t o o s l o w .

Everyone else rushed their EGAs to the market, but Orchid Technology took the time to do it right. Orchid's TurboEGA™, from the *inventors* of PC TurboProcessing, packs a high-speed Turbo and an EGA into one slot, for the world's fastest EGA.

Graphics with Speed

TurboEGA makes IBM PCs and XTs run faster than an AT. It brings dazzling speed to sluggish graphics programs. *All* types of software run faster, so you finish more quickly. Transparent to the user, you won't know it's there *until you see its speed.*

We are so confident that our TurboEGA is the ultimate graphics card that we decided to give away a *free* copy of Microsoft's Windows™ with each TurboEGA*. Even Microsoft recommends that you run Windows on an AT. Now Orchid's TurboEGA gives you AT speed so XT users can use this number one windowing software at the speed they demand.

The Complete Solution

Only Orchid puts this much performance into one slot *and* comes fully loaded with 256K of RAM, so there's no hidden cost. TurboEGA is the *complete* graphics solution.

Pick up the phone and find out how you can have the EGA the competition wishes *they* had: Orchid's TurboEGA, the world's fastest.

If you have an AT or a system with 80286 speed, ask about the Orchid EGA™—four graphic-card compatibility in one slot.

Features:

- Powered by 7.2MHz 80286
- Optional 5 or 8MHz 80287
- 256K of on-board RAM
- Four graphics modes: EGA, CGA, MDA, Hercules
- Supports the long list of EGA software:
  - Microsoft Windows & Word
  - Lotus 1-2-3 & Symphony
  - Topview
  - Framework
  - Chart-Master
  - AutoCAD
  - PC Paint Plus

  and more...

47790 Westinghouse Dr.
Fremont, CA 94536
415/490-8586
Tlx: 709289

ORCHID
The Innovative Leaders

*Offer good only on TurboEGA purchases until July 31, 1986. Please allow two weeks for delivery.

CIRCLE NO. 142 ON READER SERVICE CARD

INSIDE THE SYSTEM UNIT

The Deskpro 286 has eight expansion slots—six 16-bit slots and two 8-bit slots. In the unit tested, four of those slots (all 16-bit) were occupied. Three of the four were used by the drive controllers for the hard disk, tape, and diskette (the diskette controller also contains an asynchronous communications port and a parallel printer port).

The fourth slot contained a memory card. Bucking the trend to put more and more memory on the system board, the Deskpro 286 contains no memory whatsoever there; all of its memory resides on expansion cards. Although this may be a waste of slots, it has certain advantages. If a soldered memory chip goes bad (even the Deskpro's memory card uses one row of soldered chips), the system board does not need to be removed. Further, when higher capacity 1-megabit chips become available, a new memory board can be added rather than having to replace the entire system board.

The Deskpro 286's hard-disk drive is impressive. A 30MB Control Data model, it is encased in cast aluminum with cooling fins protruding in all directions, just like an old air-cooled Volkswagen engine. It looks sturdy enough to survive a drop out a window, although it was not subjected to that test during this review.

The power supply is a 120-volt, 4-amp model that should adequately handle any expansion cards or peripherals that are installed. (The AT has a 115-volt, 5-amp power supply). The Deskpro's power supply also has a sloping top face with a 4½-inch fan for plenty of cooling air flow. The AT's power supply contains a 3-inch fan.

The Deskpro 286 comes equipped with Compaq DOS 3.1 and GW-BASIC. For the most part, Compaq DOS is the same as PC-DOS. The only items missing from the Compaq package are the BASIC.PIF and BASICA.PIF files (TopView program information files) and VDISK.LST (the source listing of the DOS RAM-disk device driver).

Compaq DOS offers several commands not available with PC-DOS. CMPQADAP.COM adapts the system so that it can display and print characters in other languages. DISKINIT.COM is a multipurpose utility that formats disks, creates partitions on hard disks, sets the date and time, copies system files and utilities to the formatted disk, and creates AUTOEXEC.BAT and CONFIG.SYS files. SETCLOCK.COM displays and sets the date and time in the proper format and language based on the country code. TAPE.EXE performs tape operations such as backup, restore, format, and directory.

Another Compaq DOS extra is ENHDISK.SYS (for enhanced disk), which is a device driver that partitions the hard disk into multiple logical drives, each of which can be referred to by a unique drive letter. Four additional keyboard drivers (KEYBDA.COM, KEYBNO.COM, KEYBSU.COM, and KEYBSV.COM) are included in Compaq DOS to accommodate users working with foreign keyboards. According to Compaq, the information required to install these drivers is supplied when the computer is sold into the relevant European market.

The Deskpro's 30MB Control Data hard-disk drive looks sturdy enough to drop out a window, although it was not subjected to that test during this review.

TESTING

Two kinds of tests were run to check the compatibility and performance of the Deskpro 286. First, a selected set of add-on products and popular applications was installed and tested. Then the *PC Tech Journal* AT Evaluation Suite was run. This is a set of compatibility and performance tests written especially for the *PC Tech Journal* series on AT-compatible machines. (For more information about these tests, see the preceding article, "Out from the Shadow of IBM," Steven Armbrust, Ted Forgeron, and Paul Pierce, p. 52.)

The add-on products installed were an Intel Above Board AT with 4MB of memory, a Microsoft bus mouse, a Hayes Smartmodem 1200B, and an IBM Enhanced Graphics Adapter and display. Every one of these products worked with the Deskpro 286. Only Above Board showed performance differences in the default mode of the Deskpro 286. It ran with two wait states on the Deskpro 286 and only one wait state on the AT. The cause of that anomaly is discussed later in this article.

The applications tested included the graphics-intensive programs Microsoft Windows and Microsoft Word; the memory-resident programs SideKick, SuperKey, and Turbo Lightning, all from Borland; the expanded-memory program Ready!, from Living Video Text; and the communications program Smartcom II, from Hayes. Although not every feature of every program was tested, those that were tested worked as intended, including printing on a parallel printer and communicating via the asynchronous communications adapter.

The IBM AT SETUP program and the IBM Advanced Diagnostics were run on the Deskpro 286. The SETUP program worked fine, except that it forced the Compaq dual-mode monitor into its medium-resolution mode (the Compaq set-up program leaves the monitor in high-resolution mode). The Deskpro 286 passed all of the advanced diagnostic tests except for the color/graphics test (test 500), due to its dual-mode monitor and display adapter.

Table 1 compares the results of the *PC Tech Journal* compatibility and performance programs—ATBIOS, ATKEY, ATPERF, ATFLOAT, and ATDISK—with those obtained when the tests were run on an 8-MHz AT with a 30MB hard disk. Two columns are listed for the Deskpro 286: one for the default mode (8-MHz processor and 6-MHz bus) and one for the high-speed mode (8-MHz processor and bus). To compare these figures with other IBM computers, refer to table 1 in "Out from the Shadow of IBM," this issue, p.52. (The compatibility and performance tests also were run on the Compaq Portable II. See the accompanying sidebar, "The Portable II: Meeting Family Expectations.")

ATBIOS was used to check the BIOS and BIOS data areas to glean information about the BIOS and to determine whether commonly used information is stored in the same locations as in the AT. ATBIOS showed that the BIOS is manufactured by Compaq, the BIOS and its data areas are compatible with the IBM counterparts for the areas tested, and the configuration of the system is as described earlier.

ATKEY tested the keyboard for compatibility and found that the Deskpro 286 uses the same 8042 keyboard processor chip that the AT uses. As a further check, an AT keyboard was plugged into the Deskpro. Although no extensive tests were performed using this set-up, the AT keyboard worked fine during normal operation on the Deskpro. This should be good news for those who want to abandon Compaq's keyboard in favor of one from the AT.

The performance of the CPU, the math coprocessor, and the memory

MAKE THE CONNECTION . . .

Our **Connection** systems will solve your problem of trying to read and write diskettes or tapes from almost any computer system using your PC.

The **Diskette Connection** is a hardware system that enables the IBM PC or compatible to read and write most 8 inch, 5¼ inch, or 3½ inch diskettes.

With our **File Connection** software programs you can transfer data files between most computer systems, including CP/M, DEC, Honeywell, Univac, IBM 3740, S/1, S/3, S/23, S/32, S/34, S/36, and S/38.

Our **Word** and **Typesetting Connection** programs use IBM standard Document Content Architecture (DCA-RFT) to transfer document files between most word processing and typesetting systems, including Compugraphic MCS, CPT, Displaywriter, OS/6, Multiset, NBI, Quadex, Xerox, and Wang.

Our **Tape Connection** system will read and write IBM or ANSI standard ½ inch 1600 BPI magnetic tape. A full size 2400 foot tape can store a 45 MByte file and be written in 6 minutes.

Since 1982, we have supplied thousands of systems to customers around the world, including IBM, NASA, AT&T, Kodak, and General Motors.

Our specialty is conversion systems and we can provide a solution to your problem. Call us today to discuss your requirements.

This ad is one of a series featuring NASA missions. For a free poster, send us your written request.

Box 1970 Flagstaff, AZ 86002
(602) 774-5187 Telex 705609

CIRCLE NO. 153 ON READER SERVICE CARD

. . . FLAGSTAFF ENGINEERING

were tested using the ATPERF program. In the default mode, the CPU clock rate tested slower than the advertised 8-MHz rate. The math coprocessor clock rate also was slower than expected. In an 8-MHz AT, both the CPU and math processor clock rates are derived from a 16-MHz crystal. The CPU clock rate is one-half the rate of the crystal (8 MHz); the math processor rate is one-third the crystal rate (5.3 MHz). Therefore, if Compaq used a technique similar to IBM, the math coprocessor should have run at 5.3 MHz instead of the 4.2 MHz that was measured during testing. Once the Deskpro 286 was switched into high-speed mode, however, both the CPU and math coprocessor ran at their expected rates.

ATPERF was used to investigate a difference between the Deskpro 286 and the AT when using Intel's Above Board. As mentioned earlier, when the Deskpro is in its default mode (MODE SPEED=FAST), the expanded memory manager (EMM) runs with two wait states in the Deskpro, compared with only one in the AT. The ATPERF results revealed why an 8-MHz computer is not always an 8-MHz computer.

As it turns out, the Deskpro 286's expansion bus is the culprit, not Above Board. Even though the 80286 CPU runs at 8 MHz (or almost 8 MHz), the expansion bus, through which all expansion cards communicate with the CPU, runs at only 6 MHz in the default state. Thus, accessing memory on an expansion card takes just as long in 8-MHz mode as it does in 6-MHz mode. (With two wait states, four clock cycles are required: two that the CPU normally uses and two wait states; with one wait state, three clock cycles are required. Four 8-MHz clock cycles take the same time as three 6-MHz clock cycles. Therefore, an 8-MHz computer operating with two wait states is the same as a 6-MHz computer operating with one wait state.)

The Deskpro 286's own memory card has special circuitry to allow that memory to run at 8 MHz in the default mode, which explains why the Deskpro's conventional memory runs with one wait state. However, to run third-party products, such as Above Board, AST RAMpage!, or others, with the minimum number of wait states, the DOS command MODE SPEED=HIGH should be used. This cannot be switched with the keyboard sequence Ctrl-Alt-\, and the Deskpro does not give an audible indication that it is in the high mode.

When Compaq introduced the Deskpro 286, no 8-MHz IBM machine yet existed. Compaq probably reasoned that a 6-MHz bus would ensure compatibility with expansion cards built for the 6-MHz AT. So, even though it made a bus that can be switched between 6 MHz and 8 MHz, it made the 6-MHz bus the default. With an 8-MHz AT now available, however, it becomes clear that Compaq's flexibility in providing compatibility and performance has paid off. If compatibility is crucial, the bus can be left running at 6 MHz. If speed is important, a switch to an 8-MHz bus (although formerly undocumented) is possible. In addition, the time required to write to video memory (using the Deskpro's dual-mode adapter) is 8-percent less than for the AT.

Even though the 80286 CPU runs at 8 MHz (or almost 8 MHz), the Deskpro 286's expansion bus runs at only 6 MHz when it is in the default state.

ATFLOAT confirmed the poor performance of the 80287 in the Deskpro 286's default mode. In high-speed mode (MODE SPEED=HIGH), the Deskpro 286 measured up to the 8-MHz AT.

The results of ATDISK show that the performance of the Deskpro 286's hard disk is in the same league as the high-performance AT models. One interesting note is that although both IBM and Compaq advertise a 30MB drive, the Compaq disk has 36 fewer cylinders and 1,566,720 fewer bytes of storage.

In addition to compatibility and performance tests, a very important factor in judging a computer is its technical documentation. Compaq documentation has come a long way from the early days of the Compaq Portable. The manuals are now printed on heavy, high-gloss paper, and contain information that is well laid-out and easy to find. The *Operations Guide* still lacks the detailed information found in the IBM manual—for example, Compaq excludes mention of the 80287—but for the most part, the documentation is helpful and well-organized.

CONTINUING TRADITION

The Deskpro 286 is a well-built computer that continues Compaq's tradition of compatibility and added value. It is better than the AT in that it offers both 6- and 8-MHz speed on the same machine. Its video write time is faster than that on the 8-MHz AT.

However, it is unfortunate that Compaq did not make the Deskpro 286's fastest operating speed (MODE SPEED=HIGH) the default speed for the computer and, further, that it did not provide a Ctrl-Alt-\ option to switch the Deskpro 286 into the faster mode from the keyboard. Obviously, the system designers included two different 8-MHz modes in the Deskpro 286 because they were not sure whether the expansion bus on the 8-MHz IBM AT was going to run at 6 MHz or 8 MHz. By making the conservative choice of the 6-MHz bus (MODE SPEED=FAST) to be the default mode, they seem to have guessed wrong.

Nonetheless, the designers should have allowed the high-speed mode to be set with the Ctrl-Alt-\ keyboard sequence. This is a quick and easy way to switch speeds even from within a program; furthermore, it offers audible feedback on the choice. As it stands now, the only way to ensure that the Deskpro 286 will always be operating at its highest speed is by placing a MODE SPEED=HIGH command into the AUTOEXEC.BAT file.

Just as unfortunate is Compaq's decision not to document the high-speed mode in its original manuals, although, by the time this review appears in print, Compaq probably will have added the description to the documentation.

Of course, only the benefit of hindsight shows Compaq's decisions on speed selection to be faulty. It is to the company's credit, and typical of Compaq, that it chose to make the Deskpro 286 compatible with the then-current IBM AT, but included enough options to allow it to perform as fast as the new 8-MHz AT. Like other computers in the Compaq family, the Deskpro 286's reputation for compatibility and performance is warranted.

Deskpro 286
Compaq Computer Corp.
2033 FM 149
Houston, TX 77070
713/370-0670
CIRCLE 352 ON READER SERVICE CARD

Steven Armbrust, a freelance technical writer, and Ted Forgeron, software project manager for Intel Scientific Computers, will be reviewing several AT-compatible computers for this series. They are the authors of the Programmer's Reference Manual for IBM Personal Computers *to be published by Dow Jones-Irwin in October 1986.*

OFFICE AUTOMATION

Tech PC, we're out to capture your microcomputer business. No one matches our diverse line of high performance products, or our comprehensive system of support, warranty, and third party maintenance service options. Tech PC is a large Southern California computer design and manufacturing firm with computer research and design specialists, production engineers, technical assistance staff, qualified sales personnel, and a strong management and investment team to provide you with state of the art products, state of the art reliability, and state of the art economy.

PORTABLES

Tech PC Portable Computers come in four different base models to match your computing needs. Our entry level **Tech PC/XT** portable computer is designed to provide a non-obstructive profile, and has a capacity to hold two thirty megabyte hard disks, a high resolution monochrome monitor with a Hercules compatible graphics card to drive it with 800 by 400 lines resolution, and never before heard of 6 expansion slots in a portable computer. If 4.77 MHz of Intel 8088 power isn't fast enough, try our **Turbo 4.77 to 8 MHz** 8088-2 or optional NEC V-2 microprocessor based portable machine. Designed to increase speed up to 170 percent of a standard PC/XT, **the Tech PC/AT and Tech Turbo PC/AT portable units are state of the art in portable performance.** The 80286 microprocessor based units can be fitted with up to 240 megabytes of high speed voice coil hard disk technology, and with the Turbo AT's 10 MHz math co-processor socket and 10 MHz 80286 microprocessor the unit can manipulate large spreadsheets with unparalleled performance.

DESKTOPS

Tech PC Desktop Computers are designed for survival. The **XT's** leave your desktop looking comfortably empty with their slim non-obtrusive profile; The **AT's** give you a powerful presence with a clean lined sturdy chassis. Combined with a 14 inch, 800 by 400 line amber monochrome monitor or a 14 inch, 1280 by 800 black and white monochrome monitor you've got **State of the Art** staring straight back at you. If 8 MHz 80286 power isn't fast enough for you... try the Tech **Turbo PC/AT** with its 10MHz 80286 and 10 MHz 80287 math co-processor socket. The Tech Turbo PC/AT gives you the **strongest, most powerful** 80286 performance available from anyone, anywhere... available with up to 16 megabytes of memory and hard disks up to 1 gigabyte with our new optical storage disk, the Tech Personal


A time-tested storage technology also connects PC users to mainframes and minicomputers. These eight subsystems offer consistent performance and control over data storage and transfer.

Nine-track Tape Systems

ROGER ADDELSON

The nine-track, one-half-inch tape subsystem has provided a standard, reliable interchange medium for more than 30 years. Although its acceptance is practically universal, some of the old protocols and standards have become obscure in today's computer environments. Other protocols have evolved to take their places.

Densities have matured from 200 bits of data per inch (bpi) through 556, 800, 1,600, 3,200, to the 6,250 bpi seen today. Tapes once held four tracks, then seven, and finally emerged with the nine-track standard now in use. One byte is represented by eight bits plus a parity bit, which are stored in parallel fashion with one bit on each of the nine tracks of the tape.

Reviewed in this article are nine-track tape subsystems from eight companies: Alloy Computer Products, Inc., Catamount Corporation, Cipher Data Products, Inc., Digi-Data Corporation, Emerald Systems Corporation, Flagstaff Engineering, Innovative Data Technology, and Overland Data, Inc.

TECHNOLOGY IN COMMON

These tape devices usually record nine tracks of data in parallel along the tape using a separate head coil for each track. Many tape units use separate read, write, and erase head coils so that the tape can be written, read for validation, and erased (if an error is detected) without stopping and backspacing the tape. A head that uses separate coils for read and write operations is called a dual-gap head. An additional erase head positioned after the read head is called a downstream-erase head.

Older 200-, 556-, and 800-bpi tape units implemented a standard format called NRZI (nonreturn to zero invert), which records a logical 1 as a flux reversal in the magnetic field and 0 as an absence of flux reversal. The mechanical requirements for this type of drive are more critical than for the high-density 1,600- and 3,200-bpi recorders that apply a PE (phase-encoded) format. PE format begins a logical 1 with a flux reversal from the nonerased direction to the erased direction and a logical 0 as a flux reversal from the erased to the nonerased. To ensure that the flux is in the proper direction in order to accomplish a reversal, a second flux reversal occurs for each bit that is recorded when the next bit is the same value, thus resetting the direction so it may again be reversed.

Most mainframe tape systems are capable of reading and writing 1,600-bpi tape, but will not recognize 3,200-

PHOTOGRAPH • FOUR BY FIVE, INC.

bpi formats. For 6,250-bpi tapes, the standard format is GCR (group-coded records), which records bits like the NRZI format does. It uses a translation table to map "artificial" bit patterns (five bits recorded for every four actual data bits) to eliminate bit patterns that yield more than two 0 bits in a row and more than one 0 bit starting or ending a subgroup. GCR also further expands every seven bits to eight using a sophisticated polynomial algorithm. The 6,250-bpi GCR format requires sophisticated electronics and is much more expensive (by about $5,000) than 1,600-bpi PE systems. The tape subsystems that are considered in this review fall within the 1,600-bpi PE category.

Data are written to the tape in blocks, with a gap of .5 to 1.5 inches of erased or unrecorded tape between each block. This gap (normally .6 inch) is called an interblock or interrecord gap (IRG). Besides separating blocks, the IRG also functions to allow for data correction. When write errors occur (which are usually caused by a crease in the tape or an otherwise bad portion of the media), the bad section of tape is erased, and thus becomes an extended IRG. The block is written again further down on the tape.

The blocks of data on unlabeled tapes are separated into files with *tape marks*. A tape mark, or file mark, is recorded by writing an erase gap of roughly 3.5 inches, followed by several hundred high-frequency flux reversals at the maximum frequency allowed by the format (NRZI, PE, or GCR). This gap and the flux reversals are located on tracks 1, 2, 4, 5, 7, and 8, while tracks 3, 6, and 9 are held constant in the erased direction. This sequence cannot be mistaken for any binary data on the tape and thus is a place holder instantly recognized by the transport. Labeled tapes have additional header and trailer records surrounding the data blocks representing a file. (See figures 1 and 2).

IRGs require space on the tape. Because an IRG occurs between each block, it is prudent to have more than one record per block. For example, if a 1,600-bpi PE tape has one 80-byte record per block, the record would consume $80/1{,}600$ (.05) inch of tape. Adding the .6-inch IRGs, a foot of tape would hold only about 22 records. If, however, the records were combined into blocks of 40 records (3,200 bytes each), then about 200 records would fit on a foot. This format is called *fixed* blocking. (Some variable-length record blocking techniques use a prefix that contains a count and record length, but these formats are not supported on most microcomputer tape subsystems.)

Two universally accepted ANSI-standard physical tape formats (labeled and unlabeled) are available, as well as an IBM/OS format. All are independent of density and recording method.

The standard for labeled tapes (figure 1) specifies 80-byte volume and header labels, followed by a tape mark, then the data blocks separated by IRGs, another tape mark and optional end-of-file (EOF) records, followed by a tape mark. The volume label is always at the beginning of a labeled tape. The record contains the six-character volume identifer and is created when the tape is initialized. The tape is then used repeatedly without reinitializing the volume label. ANSI and IBM tape labels are similar except that ANSI labels are in ASCII and IBM labels are in EBCDIC.

The ANSI standards for unlabeled interchange tapes consist of a series of data blocks, separated by IRGs, followed by a tape mark (figure 2). This sequence is repeated for each file on the tape. Some unlabeled tapes have a tape mark at the beginning of the tape just as labeled tapes have a volume label. Two tape marks occur after the last file on the tape; these two marks together specify the logical end of tape (EOT).

These tape subsystems have mainframe or minicomputer transport drives with interface cards and software for microcomputers.

The speed of a tape system in general is dependent upon several factors: data density in bpi; transport speed, which varies from 25 to 100 inches per second (ips); and transport type. In addition to the actual tape unit speed, the speed at which data are transferred to the PC is also an important factor. For 100 ips at 1,600 bpi, 160 Kbps are available for transfer.

The most common method in which an interface processes data at this speed is via one of the PC's DMA channels, which allow a large block of data to be transferred directly to memory without the constant involvement of the CPU. The chip executes one command to begin the transfer, and then is free to handle other tasks. The PC has four DMA channels available. Channel 0 is reserved for RAM refresh, and, conventionally, channel 1 is used by LANs, channel 2 by floppy-disk controllers, and channel 3 by hard-disk controllers. For the tape interface board, it is prudent to use a DMA channel that is not being used by another device during the transfer. Because two devices cannot use the same DMA channel at the same time, transfers are significantly slowed when the tape unit shares a channel. Therefore, determine the primary device to and from which data will be transferred, and choose a DMA channel for the tape interface that will not conflict with that device. Every system reviewed except one (the IDT package) offers the very useful feature of permitting the board and software to be configured for the DMA channel and I/O address the user chooses.

Surprisingly, most of these tape subsystems work more slowly on a PC/AT than on a PC or PC/XT. The problem lies in the way DMA requests are resolved: the AT system board contains two DMA controllers—one for byte transfers, one for word transfers. The byte controller is slaved to the word controller in such a way that *both* are involved in a request for byte-wide DMA. This additional hardware arbitration for control of the bus does not always allow operation at 100 ips (a typical high speed) on the AT without its dropping an occasional byte, whereas the PC and XT, which have only a single 8237A DMA controller chip, can operate a tape system at that speed safely.

TAPE OPTIONS

All of these tape subsystems consist of mainframe or minicomputer transport drives with interface cards and software developed for microcomputers. They use drives with the industry-standard Pertec/Cipher formatter protocol. The formatter is the microprocessor-based hardware interface between the host controller and the physical tape transport motors and tape heads. It is nearly always located within the tape transport itself, and provides a standard interface to diverse transport mechanisms, in the same fashion that virtually all hard-disk drives today use the ST-506 interface originated by Shugart.

. The tape controller or interface board in the PC is typically connected to the transport formatter with two 50-conductor cables. Several of the manufacturers use the same physical tape transport, and differ only in their interface board and software design. Two of the companies represented in this re-

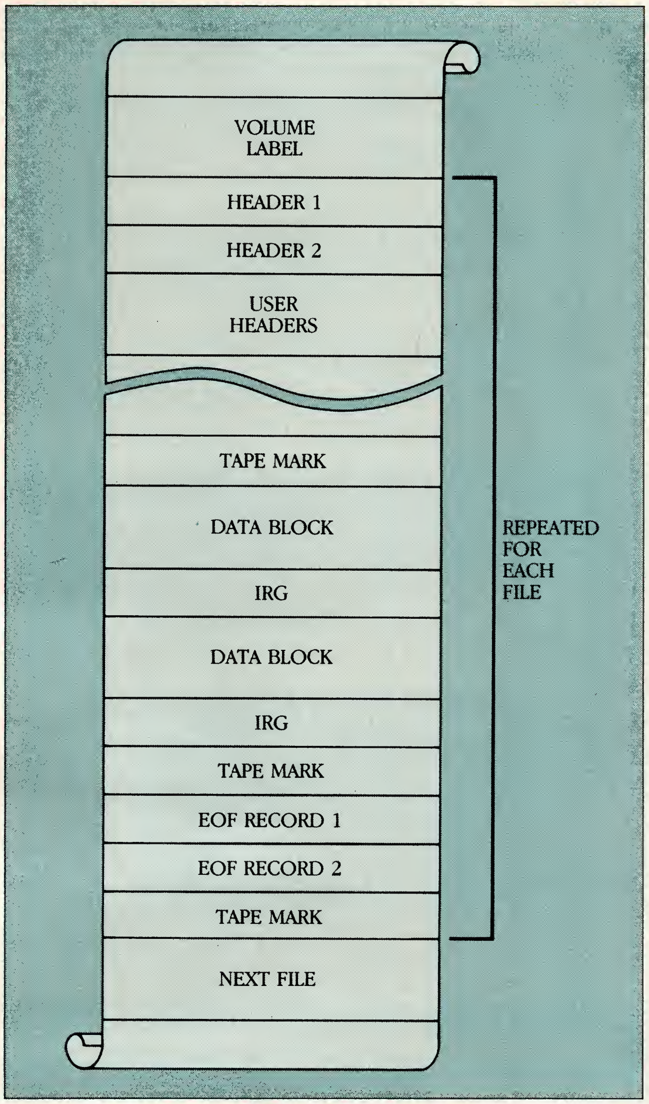
PHOTOS: *Six Varieties of Nine-track Tape Transport*

The Qualstar 1043 (top photo) was the smallest transport tested, but it works very speedily because it is equipped with an efficient set of driver software. Cipher Data's packaging of its own F880 transport (center) features a lift-top case. Alloy Computer Products and Emerald Systems both use the Pertec FS1511-19 transport (bottom). The Cipher and Pertec transports both load from the front.

Digi-Data's 2010-T tabletop transport (top photo) is small and moves tape very quickly. It also can be purchased in a rack-mounted version. Catamount Corporation packages the Cipher F880 transport (center) with its PCT-9S system. Catamount supplies a case that slides off to the rear. Innovative Data Technology furnishes a TOMACO TM-S transport (bottom) with its LEO tape system.

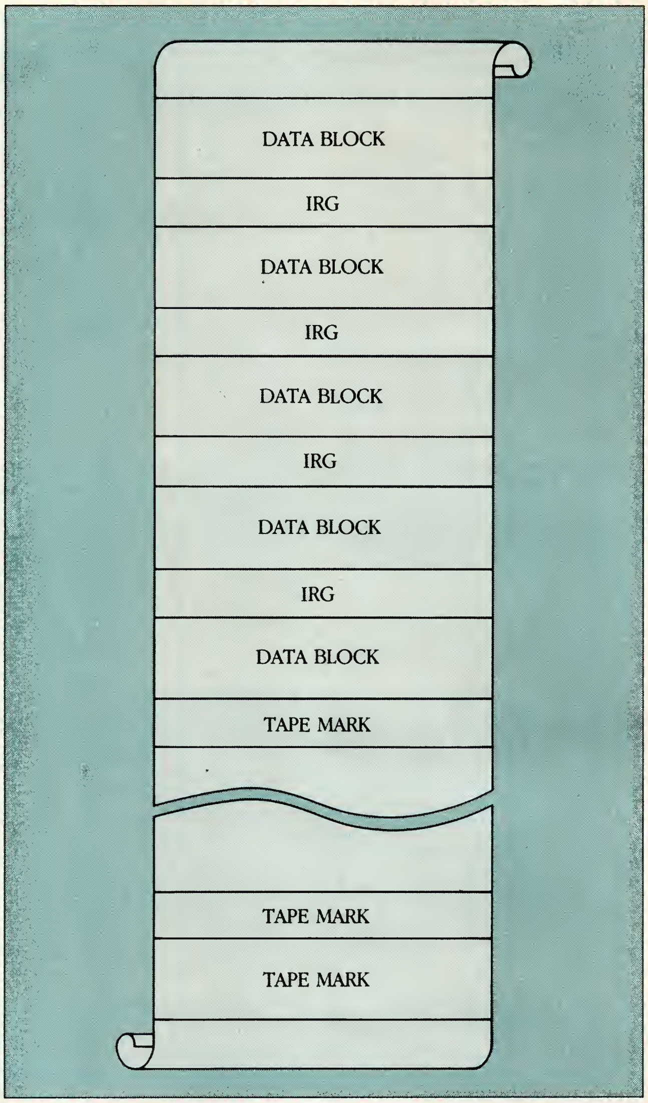
PHOTOGRAPHS • CHRISTINE ARMSTRONG / BLAKESLEE-LANE

FIGURE 1: *Labeled Tape Format*



Header and volume label information, typically used only by mainframe JCL and applications for tape control purposes, cannot be read by some of these systems.

FIGURE 2: *Unlabeled Tape Format*



Files on unlabeled tapes have no headers or labels, and tape marks serve to separate files. Two adjacent tape marks with no intervening data or IRGs are recognized as logical EOT.

view, Cipher and Catamount, use identical transports, controllers, and software, and offer essentially the same product. (See table 1 for a comparison of product features and table 2 for controller specifications. Photographs of each type of system are shown on page 97.)

Each system was evaluated on ease of installation, quality of documentation, diagnostics, level of control over the tape hardware, and, of course, benchmark time tests. (Note that the order in which the products are reviewed does not reflect any judgment made as to their quality.) For the purposes of this analysis, both labeled and unlabeled test tapes were developed containing ASCII and EBCDIC files of varying lengths and types. The EBCDIC tapes were created on an IBM 3081D under MVS, the ASCII tapes on a PDP-11/44 under UNIX. The tapes were read by the systems under examination, and written to another tape to be read by the host IBM and PDP-11. The results of these read/write operations were scrutinized closely to ensure that all data were read and written correctly.

Finally, two benchmark timing tests were run (see table 3). The first involves reading a 256KB ASCII file from tape to a DOS file on drive A: of a 4.77-MHz IBM PC. This test is a good indicator of the speed with which data can be transferred between the controller and a tape, and of the efficiency of a controller's DMA facility. The other test times the process of skipping 1,000 tape marks. Intervening data were not read, so this second test is an indicator of the average speed with which selected files can be located on a large tape containing many files—it can be thought of as a test of the system "seek" time.

Flagstaff Engineering. The Flagstaff controller card and software (included in its model 8820 system) were evaluated with the Qualstar 1043 transport.

Flagstaff's installation instructions were excellent and installation went fairly well. It is, however, a bit awkward to pull the two 50-pin ribbon cables through the back of the PC, attach them both to the board, and then try to seat the board on the bus. A suggestion implemented by several other manufacturers is to combine the cables at the PC end into one large DB-style connector, and plug it into the card externally on the PC backplane. Software installa-

TABLE 1: *Tape System Features*

	ALLOY COMPUTER PRODUCTS	CATAMOUNT CORP.	CIPHER DATA PRODUCTS	DIGI-DATA CORP.	EMERALD SYSTEMS CORP.	FLAGSTAFF ENGINEERING	INNOVATIVE DATA TECHNOLOGY	OVERLAND DATA
PRICE	$6,595	$4,950	$5,995	$3,995	$7,295	$3,495	$4,295	$3,795
LANGUAGE INTERFACE	●	●	●	●	○	○	●	●
LAN SUPPORT	●	○	○	○	●	○	○	○
DIAGNOSTICS	○	○	○	●	●	●	●	●
LABEL SUPPORT								
INIT	○	○	○	○	○	●	○	○
READ	○	○	○	○	●	●	●	○
WRT	○	○	○	○	○	●	○	○

● = *Yes*
○ = *No*

Full tape label support (which is provided only by the Flagstaff Engineering subsystem) is less important in a PC than in a mainframe environment. Tape label support is built into much mainframe system software and many applications; however, no such support exists in DOS, for which tape is still an infrequently used storage medium.

tion went much easier, requiring only that the FLAGIO.SYS device driver be added to CONFIG.SYS and that CONFIG.SYS have at least four buffers defined. To change the DMA channel or I/O address (interrupt channels are not used), only the jumpers on the controller board need to be changed and a parameter list added after the line DEVICE = FLAGIO.SYS in CONFIG.SYS.

Flagstaff's package comes with a diagnostic routine called TAPETEST. With a scratch (test) tape on the transport, the user brings the transport on-line and runs the routine, which writes 50 blocks of 10,000 bytes of data to the tape, rewinds, then reads back the data to verify the system operation. If any errors occur, the block in which they develop will be displayed on the screen.

The nine-track tape utility software is also well documented. It includes a menu-driven utility that processes both labeled (IBM and ANSI format) and unlabeled tapes. Output to tape supports fixed-length, variable-length, and string records; input from tape supports only fixed-length and string records. The utilities allow the skipping of files, of records within files, and of characters within records. The software provides for definition of logical record length, record terminator, and blocking factor (records per block) or block size. It also supports ASCII/EBCDIC conversion.

Flagstaff's utilities provide many convenient functions, including listing data sets on labeled tapes, initializing labeled tapes, and position commands such as file forward, file backward, block forward, block backward, go to EOT, rewind, rewind and unload, write tape mark, and erase to EOT. The tape dump utility displays the tape contents on the screen in EBCDIC, ASCII, and/or HEX; the tape duplication utility duplicates tapes between two daisy-chained transports. (*Daisy-chained* transports are placed in parallel and selected by way of a code set by jumpers within the transport.) Flagstaff also provides control over tape speed (25 to 100 ips), although high speed is not supported on the AT (because of the problem discussed above). Other controllable parameters include IRG length, density (1,600 or 3,200 bpi), and buffer size.

An important feature of nine-track tape systems is the ability to do backup and restore operations between tape and the PC's fixed disk. Flagstaff's TAPEBKR utility provides the standard functions found in one-fourth-inch streaming tape backup systems. It allows wild cards and archival select (which backs up only files that are new or modified since the last backup). It permits the user to append additional files to existing backup tapes. The software provides a DIR function for files on tape and the same control functions (speed, density, positioning) as the regular nine-track utility.

The software performed well, with two exceptions. First, it will direct the transport to skip only 99 tape marks in one operation. A tape that has a great many small files (see the benchmark test that skips 1,000 tape marks) will require multiple skip operations. This, is turn, makes it cumbersome and time consuming to position the tape at the correct point (as is reflected in the benchmark results).

The second problem was considerably more serious. During the read-from-tape-to-disk benchmark test, the final block of one test file on the tape was smaller than the block size because it contained fewer records than defined by the blocking factor. The software failed to transfer this block to the disk file. In a production job, any data contained in this final short block would have been omitted from the disk output file. In every other aspect, however, the software performed acceptably.

Overland Data, Inc. Like Flagstaff's tape controller card, the TC-50M included with Overland Data's model 1052 tape system uses the Pertec/Cipher formatter standard and can interface with most industry-standard formatted tape drives. The card and software can read both PE and NRZI format at 800, 1,600, 3,200, and 6,250 bpi, although the tape transport provided for testing, the Qualstar 1043, runs only at 1,600 and 3,200 bpi.

The installation portion of Overland's documentation is excellent, with step-by-step instructions and diagrams. The Overland package also supports the system's use with IBM XENIX System 3 and SCO XENIX System V. The XENIX utilities are included in the package.

Overland's package includes the BTEST diagnostic program, which, when used with a scratch tape, tests the write data, write file mark, erase fixed length, read data with verify, read file mark, space forward block, and rewind functions. On dual-speed transports, it tests high- and low-speed operation.

This package has a complete set of tape control functions in its DEPOT (data exchange program with optional translation) utility. DEPOT allows data interchange from tape to disk and disk to tape, ASCII/EBCDIC conversion, and

TABLE 2: *Controller Specifications*

	ALLOY COMPUTER PRODUCTS	CIPHER DATA PRODUCTS[a]	EMERALD SYSTEMS CORP.	FLAGSTAFF ENGINEERING	INNOVATIVE DATA TECHNOLOGY	OVERLAND DATA[b]
DEFAULT DMA CHANNEL	N/A	1	1	2[c]	N/A	1
DEFAULT I/O ADDRESS	300H	220H	300H	350H	330H	36CH
DEFAULT IRQ	N/A	3	2	N/A	N/A	N/A
CHANGE DEFAULTS	Only I/O address	●	●	●	○	●
CODINGS						
NRZI	●	○	○	○	○	●
PE	●	●	●	●	●	●
GCR	○	●	○	●	●	●

[a] *This controller is also used by Catamount Corporation.* ● = *Yes* ○ = *No*
[b] *This controller is also used by Digi-Data Corporation.*
[c] *Default DMA setting for PC/XT.*

The tape transport places only magnetic flux reversals on tape; the controller determines how those flux reversals represent data. Only Overland Data's controller provides the flexibility of support for all three major encoding systems.

tape positioning to an arbitrary location prior to data read. Like Flagstaff, Overland provides control over logical record length, block factor, block size, appending to existing tape, and positioning by skipping blocks and tape marks.

Overland's TAU (tape archive utility) is designed with features similar to those of one-fourth-inch streaming tape backup units. TAU permits individual DOS file backup and restore, use of wild cards, and data encryption. It can append to existing tapes and display a directory of files on a tape. It does not, however, have an archival-select feature. Overland's BACKUP utility produces a mirror-image backup, and restores at higher speeds than file-by-file backup. BACKUP copies data byte for byte to tape. The restoration process overwrites the previous contents of the disk with the restored mirror image.

Overland includes an important feature that the Flagstaff package does not—a programming language interface subroutine that gives a user program complete control over the drive. Examples are provided in C.

The TC-50M controller and software passed all tests and performed well. The software demonstrated great flexibility and ease of use. The package's one shortcoming is that it can neither read nor initialize tape labels. The user must manually skip over the appropriate number of tape marks to find the first file on a labeled tape.

Digi-Data Corporation. Although primarily a manufacturer of tape transports, Digi-Data also offers a complete nine-track tape system that incorporates its own model 2010-T transport with Overland Data's TC-50M controller and software. (See the discussion above.)

Digi-Data's 2010-T transport is the tabletop model of its 2010 rack-mounted tape transport. The 2010-T is smaller than any other transport tested except the Qualstar 1043. The tape reel is loaded from the top of the drive, beneath a transparent lid.

A comparison of this system's performance to that of Overland Data's provides an interesting insight into the influence of a tape transport on the performance of the system as a whole. The Digi-Data tape system turned in the best performance in moving tape through the transport, as measured by its ability to skip tape marks. In reading and writing tape data, however, Digi-Data finished considerably behind Overland Data's tape system, which uses the Qualstar 1043 transport.

The transport and the Overland Data controller and software functioned as described, without difficulty.

Emerald Systems Corporation. Emerald markets its own tape controller and software bundled with a Pertec FS1511-19 tape transport system and calls the package the Emerald Series 7000 Subsystem. This subsystem can read and write both 1,600- and 3,200-bpi unlabeled tapes, but offers only limited label support for labeled tapes.

The documentation includes a well-written, illustrated section on installation. Like the other units that have large ribbon cables, the process was awkward. The default DMA (channel 1), I/O address (300H), and IRQ (2) can be changed via a DIP switch on the board. The changes are implemented in the

LOOKING FOR THE BEST VALUE IN PC/AT COMPATIBLES?

INTRODUCING THE $1495 A★STAR™

Looking for a lower priced, higher quality PC/AT compatible than our new A★Star™ is like looking for a needle in a haystack. You're more apt to come up with a case of hayfever.

A★Star gives you features you won't get from *any* other vendor. Not IBM. Not Compaq. Not even those foreign manufacturers. Features like a 220 watt power supply, eight *available* expansion slots and "network ready" multi-user operation. All for only $1495. And that price includes a 1.2MB diskette drive, 512KB memory, a fixed disk/diskette drive controller, a tactile feedback keyboard (you'll love it!) and a 6/8MHz switch/software selectable 80286 CPU.

Best of all, A★Star is quality built right here in America by Wells American Corporation - an American Stock Exchange manufacturer. So now you can buy with confidence from a vendor you can trust, just like you've trusted IBM. And you'll get a genuine, top quality product (not a cheap imitation) for a price even less than those "questionable" mail order clones. And if that's not enough, it's all money-back guaranteed!

The $1495 Wells American A★Star.™

A PC/AT FOR ONLY $1495? THAT'S NOTHING TO SNEEZE AT!

[Hurry! This offer *is* limited.]

☐ I'm tired of sneezing! Have someone call me immediately to take my order.

☐ A★Star sounds terrific. Tell me more.

Name:__________

Company:__________

Phone:__________

Address:__________

City:__________

State:__________ Zip:__________

Wells American

803/796-7800•TWX 510-601-2645

Sunset Boulevard • West Columbia, SC 29169

CIRCLE NO. 124 ON READER SERVICE CARD

TABLE 3: *Benchmark Timings*

	ALLOY COMPUTER PRODUCTS	CATAMOUNT CORP.	CIPHER DATA PRODUCTS	DIGI-DATA CORP.	EMERALD SYSTEMS CORP.	FLAGSTAFF ENGINEERING	INNOVATIVE DATA TECHNOLOGY	OVERLAND DATA
CONTROLLER MANUFACTURER	Alloy Computer	Cipher Data Prod.	Cipher Data Prod.	Overland Data	Emerald Systems	Flagstaff Engineering	Innovative Data Tech.	Overland Data
TRANSPORT DEVICE	Pertec FS1511-19	Cipher F880	Cipher F880	Digi-Data	Pertec FS1511-19	Qualstar 1043	TOMACO TM-S B	Qualstar 1043
SKIP 1,000 MARKS[a]	11:40	5:00	4:59	4:02	16:08	52:32[b]	4:30	4:23
256KB FILE READ[a]	1:49	2:02	1:59	2:11	2:23	2:08	1:55	1:25

All timings in minutes: seconds. The configuration used was a dual-floppy-disk 640KB PC running a 4.77-MHz 8088.
[a] *(100 ips, 1,600 bpi).*
[b] *The software can skip only 99 tape marks in one operation. This represents an average of five reads of 99 tape marks times 10.*

Data read performance of a tape subsystem is constrained somewhat by standard tape speeds and densities. Speed of skipping tape marks is much more dependent on the design of the transport, as reflected by the considerable variation in times.

software through a parameter list when calling the program.

The diagnostics program, TTU (tape test utility) is the most comprehensive of any tape system tested. It checks all features, such as setting density, rewind and unloading, erasing, searching, positioning, reading, writing and speed selection, all under user control. The diagnostic portion of TTU includes a quick test, comprehensive test, nondestructive read test, and write test. These tests check for the on-line, drive ready, and Pertec/Cipher formatter ready signals; check to see that the tape has reached the load point; write different patterns until EOT is encountered; rewind the tape; read the written blocks; and then read the blocks in the reverse direction. The diagnostics display the status of 30 different functions on the screen as the test proceeds.

The tape utility software TIE (tape import export) does not function as well as TTU. Although it provides many of the functions found in other vendors' utility packages, it is a higher level product that gives the user less control over actual nine-track tape functions. For example, the user cannot specify a blocking factor for input from a tape; instead, TIE uses the block size found on the tape—a figure that cannot be overridden. Neither can the user specify such basic parameters as record length.

Emerald's "high-level" solution to this limitation is to provide RFS (record format specification), a utility feature that enables users to define the format of their input and output. This is nice if the data format is to be converted between tape and disk, but it is cumbersome for verbatim reading and writing of preformatted data because the user must specify an RFS format for each different record type written, even if the format does not change. The software does provide a helpful feature for reading data into a format for spreadsheets or database packages: the CSV (comma-separated variable) record. This feature, which uses RFS, writes data from a record with variables surrounded by quotes and separated by commas.

The package's support for labeled tapes is limited to RFS. If the user knows the format of the label, it can be specified under RFS; otherwise the label must be skipped. Likewise, label initialization must be performed from a DOS file to tape using an RFS specification for the label. TIE does include control over density, rewinding, block count, ASCII/EBCDIC conversion, output block size (not blocking factor), and positioning for tape marks and blocks.

An important facility offered by the Emerald and Alloy systems is the ability to run under Novell's NetWare and Advanced NetWare LANs. Emerald provides excellent, detailed instructions on determining the I/O address, interrupt level, and DMA channel to prevent conflicts with the network hardware and software, and includes specialized network installation worksheets. The tape controller is installed in a network node and not in the server itself. The tape software is network-tolerant rather than network-cooperative because no customized version is provided: Although network file servers may be backed up to nine-track tape, Novell's network security scheme is not preserved by Emerald's tape software during a restore of a file server. Alloy's network support, described below, is a great deal more comprehensive.

Emerald Systems has a comprehensive file archival system called ASP (archival storage protector). It supports two modes: streaming file-by-file and streaming track-by-track. Streaming file-by-file mode is twice as fast as conventional file-by-file backup/restores and provides a directory for file location. It allows the user to back up and restore only part of a disk's contents. Files can be selected individually, by time and date of last modification, or by wild card. Users on a Novell LAN can specify the server to back up. They also can associate a password with a backup operation; the password will be needed to initiate a restore. Emerald's nine-track ASP is essentially the same software that is included with its one-fourth-inch streaming tape backup.

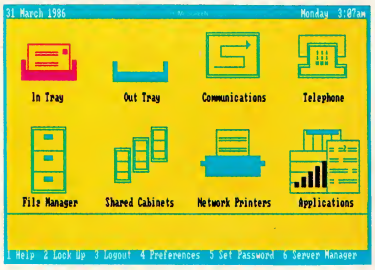
Streaming track-by-track is available for use only with an Emerald hard-disk subsystem. It provides no directories and requires a complete backup or restore for an entire disk volume. It is, however, about five times as fast as a conventional file-by-file backup/restore.

The Emerald system performed as documented. The lack of control over record size and blocking factor can be annoying. In addition, performance of the write-to-disk tests, which went well on other systems, was much more time consuming on this system. Each record had to be formatted with RFS and fitted to a predetermined block size to ensure that the time tests would not be affected by inefficient blocking. Without the exact format of the individual file headers, a tape label cannot be read. The Emerald system was tested by skipping file marks for the volume and header records on the test labeled tapes. It is unclear whether the high-level features offered by RFS are worth the restrictions on the system's versatility, especially for more technical users.

A remarkable breakthrough for LANs–one that's easy to use: Tapestry

No matter what local area network hardware you choose–your next choice should be Tapestry network operating software from Torus Systems. Here are 6 reasons why.

1 **IBM chose Tapestry,** for its overseas market, and it quickly became the hottest LAN system in Europe.



This is the friendly screen that caught Big Blue's eye.

2 **It's the world's easiest-to-use networking software**–much simpler than Novell, 3Com, or IBM because Tapestry is managed with icons. To use it, you just point to graphic symbols: the "in-tray" to get mail, the "cabinet" to access files, and so on.

You even use icons to configure the system, so almost anyone can set up a Tapestry network–and keep it running smoothly.

3 **Tapestry is the most complete LAN software money can buy.**

The basic system gives you:

- **Advanced electronic mail**

Compose memos with the Torus text editor. Send them to any person or group on the network, just by pointing to names. Track the mail you've sent. Get messages instantly.

- **Cost-effective sharing of hard disks, printers and modems**

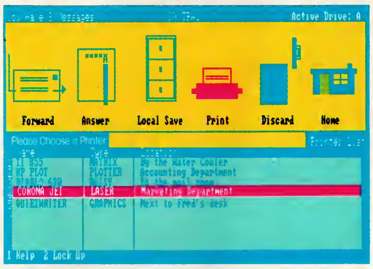
You can have up to 100 Tapestry workstation/servers and access any one of them by touching the right icon. Since all Tapestry servers are undedicated, each station is free for business as usual. You don't have to invest in specialized hardware, or sacrifice your valuable PCs.

- **Advanced communications capabilities**

Need information from the company mainframe or an outside service? Just point to the desired Service icon. Torus offers a family of network gateways* that automatically make the connection and speed your information through.

- **And there's more!**

Tapestry also provides central storage of all your applications so they too can be accessed with icons. Automatic file locking so you can safely run single-user applications not originally designed for networks. File Manager icons that let you manipulate files without using DOS commands. And a Telephone Manager that places your calls and maintains a personal electronic rolodex.



What good is a network if your people don't know how to use it? Tapestry's screens are so easy to understand, anyone can net great results without special skills or training.

4 **Tapestry supports all the standards,** like IBM Token-Ring, PC Network and 3Com hardware. Like PC-DOS 3.1 and NETBIOS. So you can choose the LAN hardware that's best for you, and you won't have to worry if your software will run.**

5 **Tapestry is proven and reliable.** Thousands of Tapestry stations have been installed in both large and small companies in the U.S. and around the world. Customers include: Proctor & Gamble, EDS, the White House, the IRS, Prime Computer, TRW, NBC News, Shell Oil, Exxon and Citicorp.

6 **With this coupon you get a FREE Network Manager AND a money back guarantee!**

Usually, Tapestry costs $495 for your first workstation (the Network Manager) and $295 for each additional station.† BUT NOW YOU CAN GET A TAPESTRY STARTER PACK (BOTH THE MANAGER AND THE SECOND STATION) FOR ONLY $295!

And we're so confident you'll love Tapestry, we'll let you return it–for any reason–within 30 days of shipment and we'll gladly refund your purchase price (less $50.00 restocking fee).

So send this coupon to us, take it to your Torus Dealer, or call in your order today, and start networking the easy way.

Name ____________

Company Name ____________

Address ____________ City ____________

State ____ Zip ____ Phone ____________

Networking Hardware ____________

Number of Workstations in Office ____________

Subtotal	$295!
Add $7.00 shipping/handling	$____
California residents add 6½% sales tax ($19.18)	$____
Amount enclosed	$____

Payment VISA MC Check

Credit Card Exp. Date ____/____

Card # ____________

Act now: offer expires August 15, 1986.

This offer is good only in U.S. and Canada.

TORUS 495 Seaport Court, Suite 105 Redwood City, CA 94063 (800) 872-5335

System Requirements: Station 1: IBM PC, XT, AT or compatible 384K, 10MB hard disk. Station 2: IBM PC, XT, AT or compatible 320K. Plus suitable networking hardware.

Notes: *ASYNC is standard, while VT100, 3270 SNA, and Remote Access are purchased separately. **Tested applications include: dBase III Plus, MultiMate Advantage, Open Systems Accounting, Displaywrite 3, WordPerfect, MS Word, Lotus 1-2-3, AutoCAD.

†6 Station Pack–$1495, 24 Station Pack–$3995.

CIRCLE NO. 184 ON READER SERVICE CARD

Alloy Computer Products, Inc. The Alloy Q-TIP system, like the Emerald system, includes Alloy's ITS-PC tape controller card, along with TIP-6 and PC 9-Track software, bundled with Pertec's FS1000 series formatter and tape transport. It can read and write PE-formatted, 1,600- and 3,200-bpi tapes.

The documentation provides step-by-step, illustrated installation instructions. This system suffers the same awkward hardware installation as the other interface cards reviewed so far. Alloy's technical support confirmed that the system does not use either interrupts or DMA, but that fact was not mentioned in the documentation. The manual does, however, show how to change the default I/O base address of 300H and transport type via DIP switches and jumpers. The Cipher Microstreamer is the default transport type. The PC 9-Track software is modified to reflect the base I/O address change with Alloy's TINSTALL program. After entry of the desired hexadecimal I/O address, TINSTALL modifies the PC 9-Track's TREAD, TWRITE, and TDUMP routines to reflect the new I/O address. The TIP-6 programs are similarly modified with Alloy's PATCH routine. It is disappointing to note that Alloy does not provide diagnostic routines.

The documentation is geared toward the end user—it is difficult to glean much technical information. Even Alloy's language interface routine, TLINK, is written for BASIC programs. The TLINK examples are excellent, but additional routines for other languages, such as Macro Assembler or C, would have been a welcome addition.

Functionally, the PC-9 Track utilities provide excellent control over the transport. Although the software cannot read labeled tape labels, the user can determine record length, blocking factor, record and block padding, record terminator, and ASCII/EBCDIC conversion. The software also affords control of tape position with commands to skip tape marks and skip records.

A strong point of Alloy's package is its support of Novell's NetWare and Advanced NetWare LANs. The tape controller can be installed in either a network node or the file server, if the server is using an Alloy hard-disk subsystem. The network installation process is well documented, and complements Novell's installation guide nicely. In addition, Alloy provides a special version of its QTIP Archival software called QTIP-Novell. Not only does it allow backup and retrieval of all files on the network, but it archives the Directory Access Masks and Directory Trustee Masks, which are necessary to preserve the network's security during file restoration. This strong compatibility is a powerful recommendation of the Alloy Tape system for use with a Novell LAN configuration.

Another of the Alloy product's assets is its TIP-6 hard-disk archival software. It has all the features of Alloy's PC-QICTAPE one-fourth-inch streaming tape backup: initialize tape, disk-to-tape backup, disk-to-tape append, tape-to-disk restoration, tape directory, and file verification (see "Moving up to Tape," Steven Armbrust and Ted Forgeron, November 1985, p. 62). However, like PC-QICTAPE, it does not support mirror-image backup/restore; it is file oriented only. The tape system does include wild cards, query (which allows a user to enter Y/N for individual file backup/restore), archival select, exclusion of files meeting user-defined criteria, and logging. It includes special commands to give the TIP-6 software specific control of the nine-track tape: set track, skip to block, skip to saveset (Alloy's term for a backup session written to tape), write end of data (EOD), and rewind. Like its little brother QICTAPE, TIP-6 has an excellent on-line help feature.

The Emerald and Alloy systems offer the important facility of running under Novell's NetWare and Advanced NetWare LANs.

The software worked as promised, affording the user moderate control over the tape transport. No major problems arose during testing.

Cipher Data Products, Inc. With Pertec, Cipher is the originator of the Pertec/Cipher formatter standards. The Cipher Series 9000 Tape Subsystem includes the Cipher F880 transport (the M990 is available as an option for reading GCR 6250-formatted tapes). The F880 uses PE formats for 1,600- and 3,200-bpi tapes.

The Cipher system was the easiest installation of all. Cipher is the only tape controller on a short-slot card: instead of large ribbon cables attaching directly to the surface of the controller card, a shielded data cable, terminated by a D-type connector, connects to the back of the card. Software installation consists of renaming Cipher's TCONFIG.SYS file to CONFIG.SYS, or editing an existing CONFIG.SYS to include Cipher's device driver. The package includes two device drivers, one for use with the PC and XT and one for the AT that suppresses high-speed data transfers and does not use interrupts.

The documentation describes Cipher's use of the DMA channels, I/O address, and interrupt (see table 2), but does not include instructions on reconfiguring the controller hardware. The manual does discuss reconfiguration of the software with the TINSTALL program. TINSTALL delivers a highly detailed, narrative instruction on reconfiguring the jumpers on the board as well as performing a reconfiguration of TDS.SYS, Cipher's device driver. Conspicuously absent from this package are diagnostic routines. Cipher suggests using its tape backup routine with a /RO command-line switch first, to test the functionality of the transport and interface before overwriting a valid backup tape with new information.

This product includes a complete hard-disk archival package, with TBACKUP and TRESTORE. DT (disk/tape) supports tape-to-disk and disk-to-tape file transfers, and TDS provides access to tape data by emulating a disk-type device and a character-type device within DOS. Utility programs are provided for positioning and erasing tapes, and performing tape dumps. Two separate utilities permit tape system access from user-written programs: BPS provides BASICA programming support for interpreter versions of BASIC, while GPTR supports most languages that link external routines via DOS LINK. DT is used to copy, and optionally reformat, data between the tape drive and a DOS device or file. It provides significant control over the transport, with commands to control block size, logical record length, record deliniator, tab expansion, data append, and ASCII/EBCDIC conversion. It provides positional control with options to skip tape marks, and skip to beginning of tape (BOT) or EOT.

Cipher's TBACKUP and TRESTORE resemble DOS BACKUP and RESTORE, with enhancements to take advantage of one-half-inch tape capabilities. TDIR provides a directory of archived files. Users may classify files to restore groups of files selectively from a tape that contains multiple backup sessions. BACKUP has been extended to allow archiving of multiple disk volumes in a single operation. DOS BACKUP's archival bit and modification date support have been retained. Reset of the archival bit on write-protected disks has

PC MAGAZINE EDITOR'S CHOICE

*...one package stands out as the best support available for Turbo Pascal programmers: **Blaise Computing's Turbo Power Tools.** This definitive set of prewritten Pascal functions and procedures will make the life of any programmer—from the beginner to the hard-core professional—easier and more productive.*

ANOTHER PLUS FROM BLAISE COMPUTING

Power Tools Plus™

Window Routines. Memory Resident Routines. Routinely.

The best just got better! Turbo POWER TOOLS, acclaimed as the best programmer support package for Turbo Pascal, now has even more functions, more detailed documentation and more sample programs.

NO SECRETS

Turbo POWER TOOLS PLUS is crafted so that the source is efficient, readable and easy to modify. We don't keep secrets! We tell you exactly how windows are managed, how interrupt service routines can be written in Turbo Pascal, and how to write memory resident programs that can even access the disk. Maybe you've heard of some undocumented DOS features that resident programs use to weave their magic. Turbo POWER TOOLS PLUS documents these features and lets you make your own magic!

Here's just part of the PLUS in Turbo POWER TOOLS PLUS:

- **WINDOWS** that are stackable, removable, with optional borders and a cursor memory.
- **FAST DIRECT VIDEO ACCESS** for efficiency.
- **SCREEN HANDLING** including multiple monitor and EGA 43-line support.
- **POP-UP MENUS** which are flexible, efficient and easy to use, giving your applications that polished look.
- **INTERRUPT SERVICE ROUTINES** that can be written in Turbo Pascal without the need for assembly language or inline code.
- **INTERVENTION CODE** lets you develop memory resident applications that can take full advantage of DOS capabilities. With simple procedure calls, you can "schedule" a Turbo Pascal procedure to execute either when a "hot key" is pressed, or at a specified time.
- **PROGRAM CONTROL ROUTINES** allow you to run other programs from Turbo Pascal, and even execute DOS commands.
- **MEMORY MANAGEMENT** allows you to monitor, allocate and free DOS-controlled memory.
- **DIRECTORY AND FILE HANDLING** support to let you take advantage of the newer features of DOS including networking.
- **STRING** procedures allowing powerful translation and conversion capabilities.
- **FULL SOURCE CODE** for all included routines, sample programs and utilities.
- **DOCUMENTATION, TECHNICAL SUPPORT** and attention to detail that have distinguished Blaise Computing over the years.

Turbo POWER TOOLS PLUS** supports Turbo Pascal Version 2.0 and later and is just **$99.95.

*Another quality product from Blaise Computing: **Turbo ASYNCH PLUS™***

A new package which provides the crucial core of hardware interrupt support needed to build applications that communicate. ASYNCH PLUS offers simultaneous buffered input and output to both COM ports at speeds up to 9600 baud. The XON/XOFF protocol is supported. Now it also includes the "XMODEM" file-transfer protocol and support for Hayes compatible modems.

The underlying functions of Turbo ASYNCH PLUS are carefully crafted in assembler for efficiency and drive the UART and programmable interrupt controller chips directly. These functions, installed as a runtime resident system, require just 3.2K bytes. The high level function are all written in Turbo Pascal in the same style and format as Turbo POWER TOOLS PLUS. All source code is included for just $99.95.

BLAISE COMPUTING INC.

2560 Ninth Street, Suite 316 Berkeley, CA 94710 (415) 540-5441

ORDER TOLL-FREE 800-227-8087

Calif. residents call (415) 540-5441

YES, send me the PLUS I need! Enclosed is $______ for ☐ Turbo POWER TOOLS PLUS ☐ Turbo ASYNCH PLUS ☐ OTHER ______. (CA residents add 6½% Sales Tax. All domestic orders add $10.00 for Federal Express shipping.)

Name: ______ Phone: (___) ______

Shipping Address: ______

City: ______ State: ____ Zip: ______

VISA or MC #: ______ Exp. Date: ______

CIRCLE NO. 104 ON READER SERVICE CARD

AT performance at an XT price. Introducing the TeleCAT-286. $2995. Complete.

With TeleVideo, you always settle for more.

For some time, you've known exactly what kind of PC you could get with a mid-range budget.

But now, you can settle for a whole lot more. With the new TeleCAT-286™ from TeleVideo.

It starts you off with IBM AT compatibility. 512K RAM. A 20MB hard disk. A 1.2MB floppy. And everything else you need. Like an Intel 80286 CPU that runs at either 6 or 8 MHz. Even a high-resolution 640 x 400 pixel monitor for text and graphics.

To make even better use of internal space, we socketed the TeleCAT-286 for one MB of RAM, and also included serial and parallel ports on the motherboard. As a result, we can still give you three extra expansion slots.

And we didn't stop there. We've also designed more ergonomic features into the TeleCAT-286. Including sculptured keycaps on a high-quality keyboard. LEDs right on top of the three critical locking keys, so they won't get covered up by overlays. And a footprint that's 28% smaller than the IBM AT's. So you get more of your desk back, too.

There's a lot more we could say about the TeleCAT-286. But it's an even better idea to get your hands on it. So call 1 (800) TELECAT, Dept. 154, and we'll tell you where you can try one.

The TeleCAT-286. Our 20MB version is $2995; 30MB, $3495. For high performance at a low price, don't settle for less.

TeleVideo®

Settle for more.

TeleVideo Systems, Inc., 1170 Morse Avenue, Sunnyvale, CA 94088-3568 • (408) 745-7760

CIRCLE NO. 182 ON READER SERVICE CARD

always been a problem; but Cipher's TBACKUP includes an option to bypass altering the bit. In addition to wild cards, it can exclude specified groups of files from backup. Mirror-image backups are not supported.

Cipher's package is complete, functions as documented, and performs well. It was one of the fastest packages tested and offered a wide array of features. For non-networked applications, this is the package of choice.

Catamount Corporation. The PCT-9S tape system evaluated for this article was found to be identical in all respects to the Cipher Series 9000 reviewed above. The tested configuration included Cipher's F880 transport, short-slot controller board, software from DataTrac, and D-connector-equipped shielded data cable. Catamount purchases its software and controller from the same sources as Cipher. In fact, the software program files were identical byte for byte to those shipped with the Cipher system. An advantage to Catamount's product is that the company can provide other transports besides the Cipher F880, including the Qualstar 1052 (with transport-specific software), the Kennedy 9600, the Cipher M990 (for 6,250-format tapes), and the Qualstar 701, a compact, lightweight unit for use with seven-inch tape reels.

In testing also, the Catamount system operated identically to the Cipher Series 9000. (Refer to table 1.)

Innovative Data Technology, Inc. IDT offers the most unique package. The model 1012 tape system includes a TOMACO TM-S transport, with the Pertec/Cipher formatter-compatible LEO tape controller and software. The device supports PE and GCR formats for densities of 1,600, 3,200 and, optionally, 6,250 bpi.

LEO is unique because it is the only controller tested that does not use the PC's DMA channels to transfer the tape data to and from memory. The company instead employs a technique that it calls "buffer swapping." The LEO controller uses two data buffers, one of which is always resident in the PC's memory, the other is in the controller card's memory. When the buffers are swapped, IDT claims that essentially a zero-time, DMA-like transfer between the controller and PC is accomplished. The company suggests that this process dramatically increases the throughput of the controller. Its claim was partially verified by the benchmark tests.

IDT's installation documentation is nonexistent. The company provides complete engineering-level specifications, and the schematic had to be read to determine where to attach the cable connectors to the transport. Again, connection to the PC was awkward because of the two ribbon cables. Although IDT alludes to an ability to reconfigure the base I/O address in the future, for now it is preset to 300H. If that conflicts with an existing device, nothing can be done. The package does include TT (tape test) diagnostic routines that perform a rather comprehensive test verifying the basic controller command functions, controller status reporting functions, formatter functions, transport movement, and basic data integrity. TT also performs data and blocking function checks, position function checks, media and read-write performance checks, and offers the ability to define custom test sequences.

As it is, the manual is virtually useless to the nontechnical user. Complex technical information is mixed in with the operational instructions. The level of documentation is good, but the technical notes should be separated from the functional instructions.

The LEO tape system includes several software utilities. The MT standard device driver provides an extension to DOS that makes the tape device look like a standard data file opened for sequential access, permitting most assemblers, compilers, and interpreters to work with the tape device. The ANSI tape utility program allows files to be transferred to and from single- and multiple-volume, industry-standard, one-half-inch nine-track magnetic tape in the format defined by ANSI X3.27-1978. Files may be transferred to and from a PC, XT, or AT under DOS 3.1 or later.

The utilities permit the user to specify DOS or ANSI path name, ASCII/EBCDIC conversion, tape positioning (BOT, EOT, FILE), block size and blocking factor. The TCMD tape command program includes 20 commands to select drive, rewind, status query, ASCII/EBCDIC conversion, speed, write both data and tape marks, space forward and reverse *n* blocks or *n* files, tape dump, tape compare, and help.

LEO provides a good high-level language interface for both C and BASIC. Many of LEO's routines include a C listings from which a programmer can easily lift sections of code. LEO uses a standard DOS device driver that is loaded at boot time. IDT's ANSI and IBM tape utilities and TCMD (tape command) program offer a wide range of tape-to-disk read/write and position options. The ANSI routines do not offer much control over blocking and deblocking or specific positioning, but

TCMD offers better positioning control. IDT's blocking and file control limitations are similar to Emerald's.

LEO's TAP (tape archival program) performs the standard tape backup, restore, and directory functions. Other features include tape initialization and an override for directory names (which allows files restored from a backed-up subdirectory to be written to a new subdirectory, so that the original files may be retained if desired). TAP is a command-line oriented program to which all parameters are passed on a single line. The backup function features selection by wild card, archival bit, and modification date.

IDT's routine performs as documented, although the ANSI tape functions are limited. The high-level language interface is good, but it has the same blocking and deblocking limitations as the utilities.

WHITHER TRACK?

All the systems tested conform closely to the standard Pertec/Cipher formatter specifications. All were essentially free of bugs once installation problems were resolved. None could in any way be thought of as unacceptable.

These systems exhibited good tape archival and restore utilities for PC fixed-disk backup, but the packages were divergent in the level of control offered over tape-to-disk and disk-to-tape transfer functions. Variations in speed also were evident, especially in the speed of tape positioning by skipping file marks. Cipher is the best overall system for non-networked applications, with Overland Data and Flagstaff Engineering coming in a close second. Only Emerald and Alloy offer systems suitable for application in networked environments. Moreover, Alloy's maintenance of file server network security through a full backup/restore cycle makes it stand out as the system of choice for users with a Novell LAN.

Although the nine-track tape may seem an anachronism when considered in a PC context, it offers important advantages as a link between the PC and the mainframe: no new technology is required at the mainframe end; it is more secure than telecommunications links because it can be sent by courier or hand carried between locations; carefully-formatted data tapes can be read by virtually any modern mainframe computer; and the tape control procedures already in place at most MIS sites facilitate control of access to tape-transmitted data. Furthermore, while nine-track tape is not as cost-effective as fixed-disk backup alone, backup ability is a no-extra-cost advantage when tape support is required for other purposes. Like COBOL, nine-track tape has been around for a long time. And, like COBOL, a strong set of standards is the key to that longevity.

Q-TIP, with PC-9: $6,595
Alloy Computer Products, Inc.
100 Pennsylvania Avenue
Framingham, MA 01701
617/875-6100
CIRCLE 344 ON READER SERVICE CARD

PCT-9S Tape Subsystem: $4,950
Catamount Corporation
2243 Agate Court
Simi Valley, CA 93065-1898
805/584-2233
CIRCLE 345 ON READER SERVICE CARD

Tape Subsystem: $5,995
Cipher Data Products, Inc.
10225 Willow Creek Road
P. O. Box 85170
San Diego, CA 92138
800/4-CIPHER; 619/578-9100
CIRCLE 346 ON READER SERVICE CARD

2000 PC: $3,995
Digi-Data Corporation
8580 Dorsey Run Road
Jessup, MD 20794
301/498-0200
CIRCLE 347 ON READER SERVICE CARD

Series MTM80-7000: $7,295
Emerald Systems Corporation
4757 Morena Blvd.
San Diego, CA 92117
619/270-1994
CIRCLE 348 ON READER SERVICE CARD

Model 8820: $3,495
Flagstaff Engineering
1120 Kaibab Lane
Flagstaff, AZ 86001
602/774-5188
CIRCLE 349 ON READER SERVICE CARD

Model 1012: $4,295
Innovative Data Technology
5340 Eastgate Mall
San Diego, CA 92121
619/587-0555
CIRCLE 350 ON READER SERVICE CARD

Model 1052: $3,795
Overland Data, Inc.
5644 Kearny Mesa Road
San Diego, CA 92111
619/571-5555
CIRCLE 351 ON READER SERVICE CARD

Roger Addelson is the assistant director of academic computing at Loyola University of Chicago. *The author gratefully acknowledges the assistance of Joseph Kaplenk in the testing of these tape subsystems.*

Smooth Curves

Connecting a set of points with a smooth line requires special techniques. This procedure imposes an intuitive polynomial interpolation to achieve smooth curves.

MICHAEL A. COVINGTON

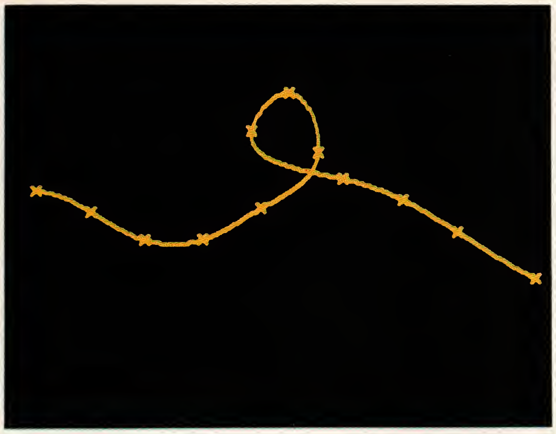
Drawing a smooth curve that connects an arbitrary set of points is a classic problem in computer graphics. It arises frequently in picture drawing—especially from data entered with a mouse or graphics pad—and in drawing graphs from numerical data. An algorithm that produces smooth curves also provides a way to eliminate the raggedness that results from magnifying a picture composed of a limited number of pixels. This, in turn, makes it possible to draw characters or symbols from very concisely stored descriptions; each character can be summarized by just a few points if the computer can connect them appropriately.

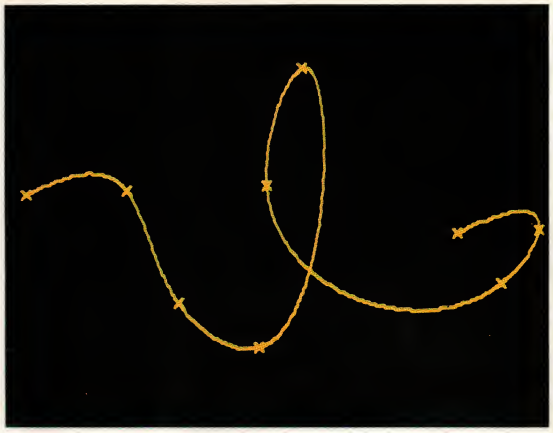
Using the Turbo Pascal program provided here, SMOOTH.PAS (listing 1), a user can mark points on the screen that the program then connects with a smooth curve. The curve passes through the points in the order in which they were marked, crossing over itself as many times as necessary to do so. This curve-fitting procedure can be adapted for use in other programs. The computation does not rely on features specific to Pascal so translation into other languages should be easy.

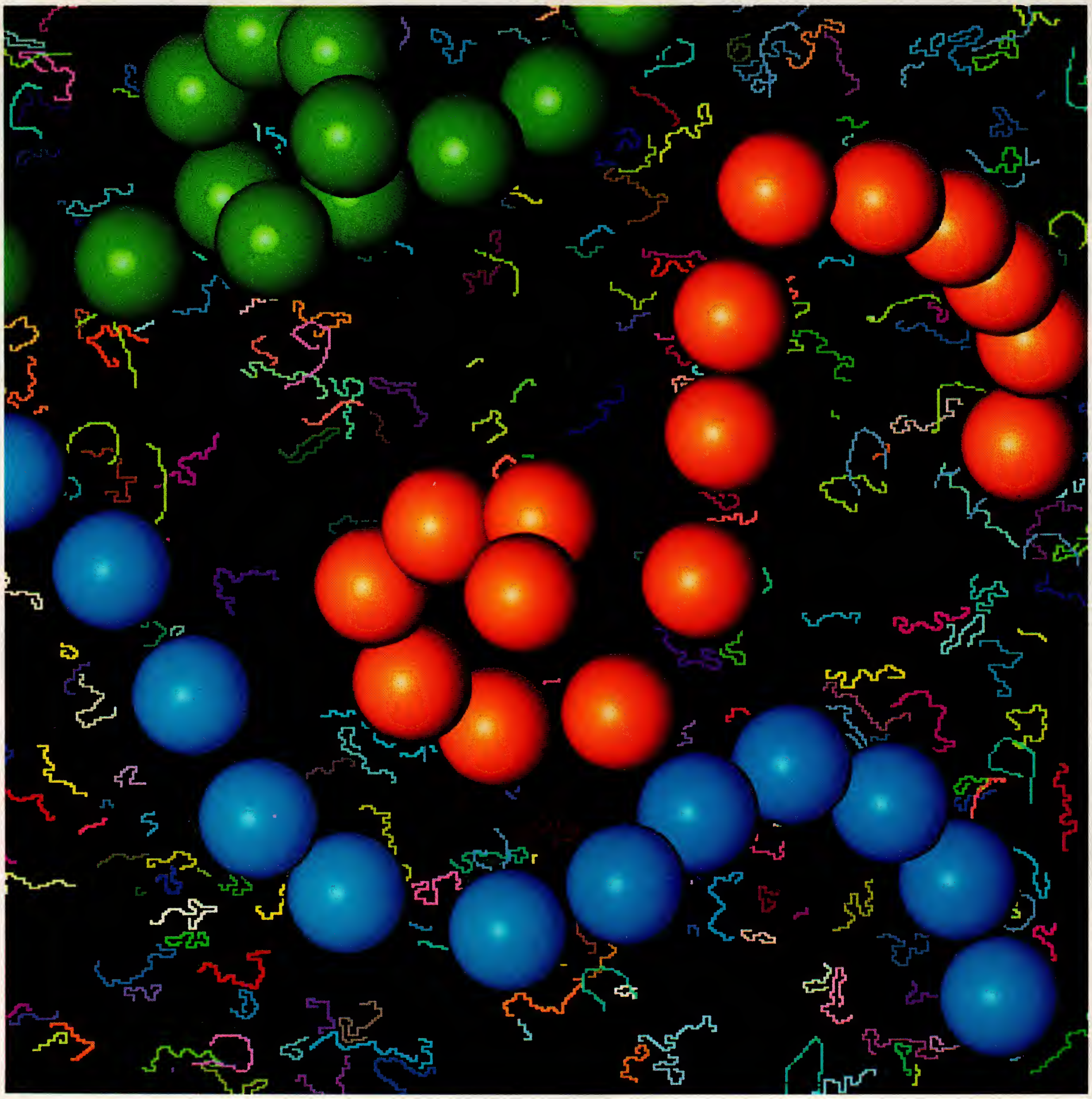
Users should note that this program was written for use with the IBM Color Graphics Adapter (CGA); making it run well from an IBM Enhanced Graphics Adapter (EGA) requires some modifications. It can be made to run without the unexpected yellow background by taking the comments out of the invocation of **graphbackground(black)** in the two places it occurs. The EGA does not respond in the same way as the CGA to Turbo Pascal's built-in **hirescolor** procedure. On the EGA, the **hirescolor** specified becomes the background color unless **graphbackground** also is specified, and the default foreground color is green. This is because Turbo Pascal uses the CGA in a mode that the EGA does not emulate.

Mathematically speaking, the problem of smooth curve creation is actually one of *interpolation*. The position of

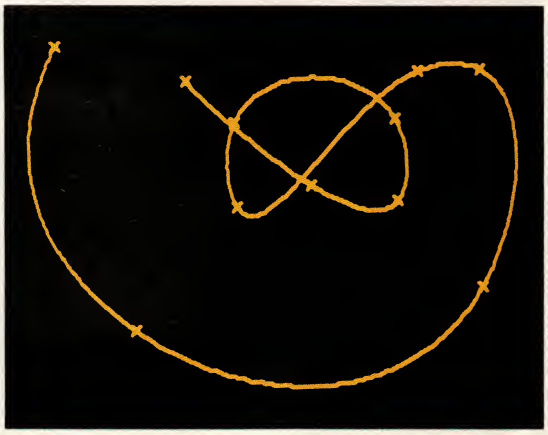
The screen shots below show examples of the smooth curves obtainable using the program presented here. The points are joined in the order they are entered.

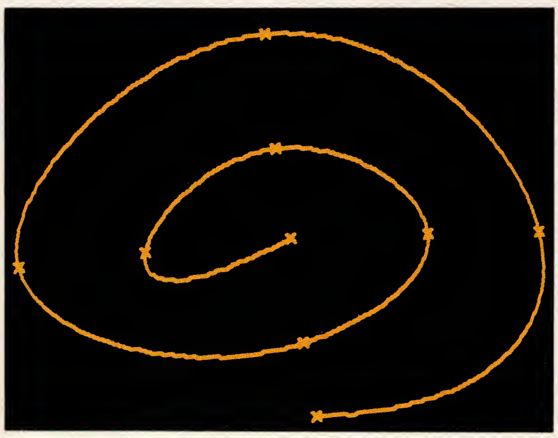


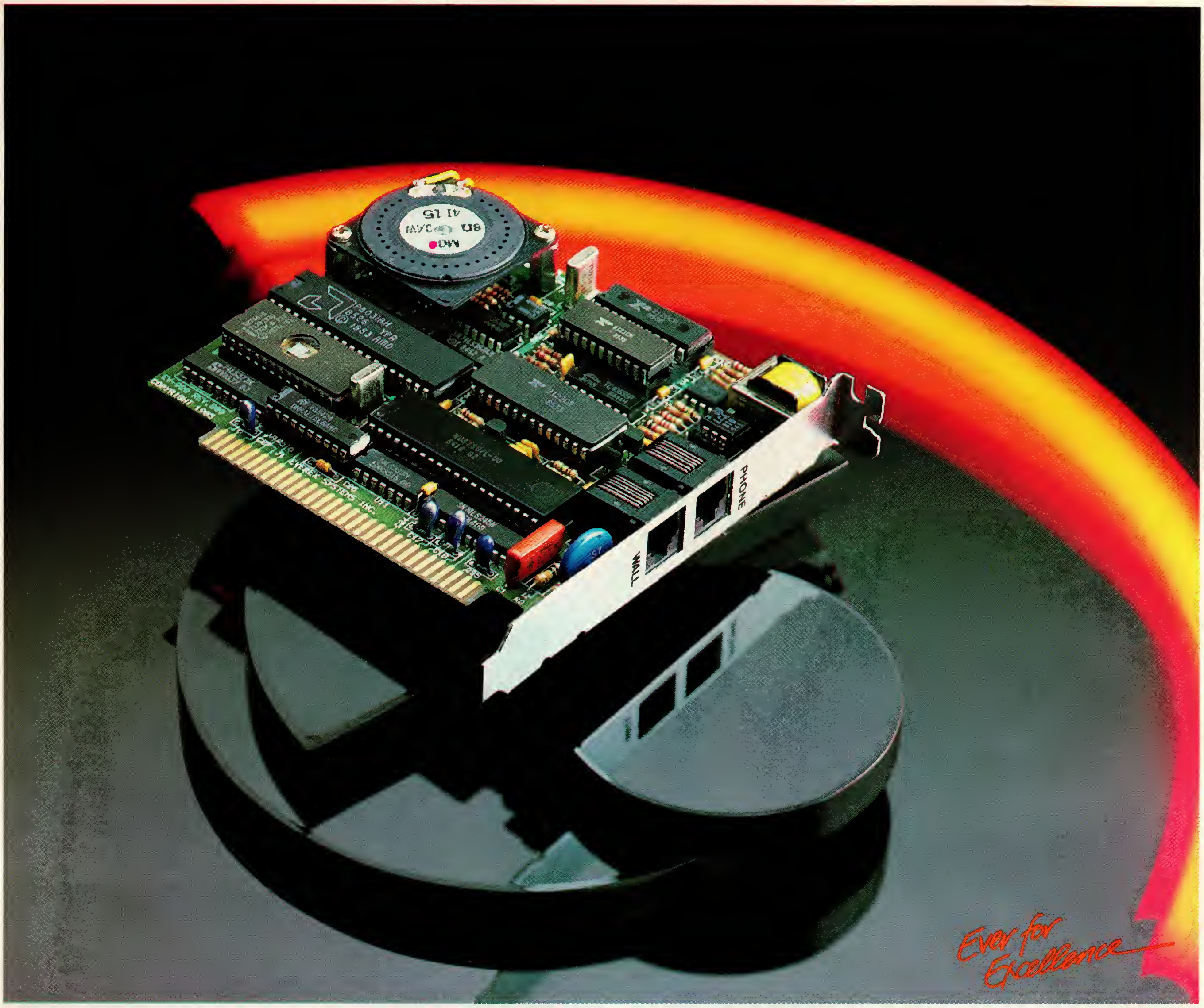




COMPUTER GRAPHIC • NATALE EAST







Evercom™... 1200 baud, $249

Now the EVEREX Evercom modem is available on a half-size card, perfect for the short slots in the IBM XT, Portable, and many compatibles. EVEREX has combined innovative engineering and technology to produce a quality Hayes-compatible 1200 baud modem that out performs the competition at a lower price. EVEREX engineering is the key ingredient that delivers QUALITY and FEATURES without sacrificing PRICE.

Before you buy a modem, check these features:

	EVEREX EVERCOM	***HAYES*** NEW 1200B
• Automatic dialing, answering, and redialing	***YES***	YES
• Half-size card	***YES***	YES
• Call progress monitoring	***YES***	YES
• Works in short slot of XT	***YES***	NO
• Automatic data-to-voice transition	***YES***	NO
• Reports serial port speed mismatch	***YES***	NO
• Detects receiver off-hook	***YES***	NO
• Configurable from COM1 - COM4	***YES***	NO
• Supports 132 columns	***YES***	NO
• Communications software included	***YES***	$80 extra
• ***List Price***	***$249***	$489

Compatibility Guaranteed!!

The Evercom short card modem is guaranteed to be compatible with the Hayes 1200B or your money back for the first 30 day evaluation period.

Call today for dealer pricing and start getting EVEREX VALUE... QUALITY and FEATURES at the LOWEST POSSIBLE PRICE. Why settle for less?

1-800-821-0806

in California

1-800-821-0807

EVEREX

EVER for EXcellence

48431 Milmont Dr. Fremont CA 94539
(415) 498-1111

CIRCLE NO. 169 ON READER SERVICE CARD

each point on the curve is described by two coordinates, x (the column number) and y (the row number). For some points, both x and y are given. A method is needed to connect the points so that for any x, an appropriate value of y can be computed, making it possible to fill in the whole curve.

The interpolation function must pass through all the known points and behave reasonably between them. It also must be easy to compute. Some mathematical functions, such as sines and logarithms, take a long time to evaluate on a conventional computer. As far as possible, it is best to avoid these functions and use only addition, subtraction, multiplication, and division.

This suggests that the interpolating function should be a polynomial, an equation of the form

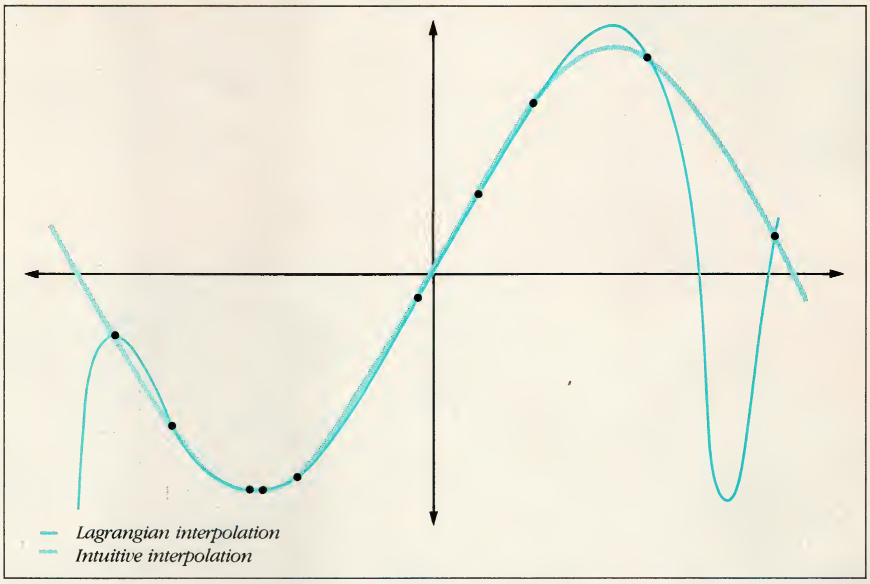
$$y = ax^n + bx^{n-1} + ... + cx^2 + dx + e$$

where a, b, c, d, and e (the coefficients) are constants, and n is used to represent the degree of the polynomial.

The degree determines the complexity of the polynomial's curve: the maximum number of bends in the curve is equal to the degree minus 1. A polynomial of degree 1 (a linear equation) describes a line (with no bends) and therefore can be fitted to any two points. A quadratic, the degree of which is 2, describes a parabola (that has one bend) and can be fitted to any three points. A cubic, of degree 3, describes a curve that bends twice and can be fitted to any four marked points.

In general, for any k points, a polynomial of degree $k-1$ exists that passes through all of them; using this polynomial is known as *Lagrangian interpolation*. The polynomial that passes through all the points may behave very oddly between them. In figure 1, a ninth-degree polynomial is fitted to a set of ten points. The hairline curve represents the Lagrangian interpolation. Near the middle, the curve is not bad, but at the ends, it differs wildly from an intuitive idea (as shown in the tint line) of what a smooth curve connecting the points should look like. The Lagrangian interpolation approach, then, does not produce the desired result here.

FIGURE 1: *Polynomial Interpolation between 10 Points*



Lagrangian interpolation (the hairline curve) does not produce curves that are intuitively satisfying. Here, a ninth-degree polynomial is fitted to a set of 10 points. The curve passes through all points, but its behavior between them is not what the arrangement of the points would suggest. The same set of 10 points are linked (with the tint line) by an intuitive interpolation that might have been drawn freehand. This is the kind of curve that a good curve-fitting program should generate.

SPLINE CURVES

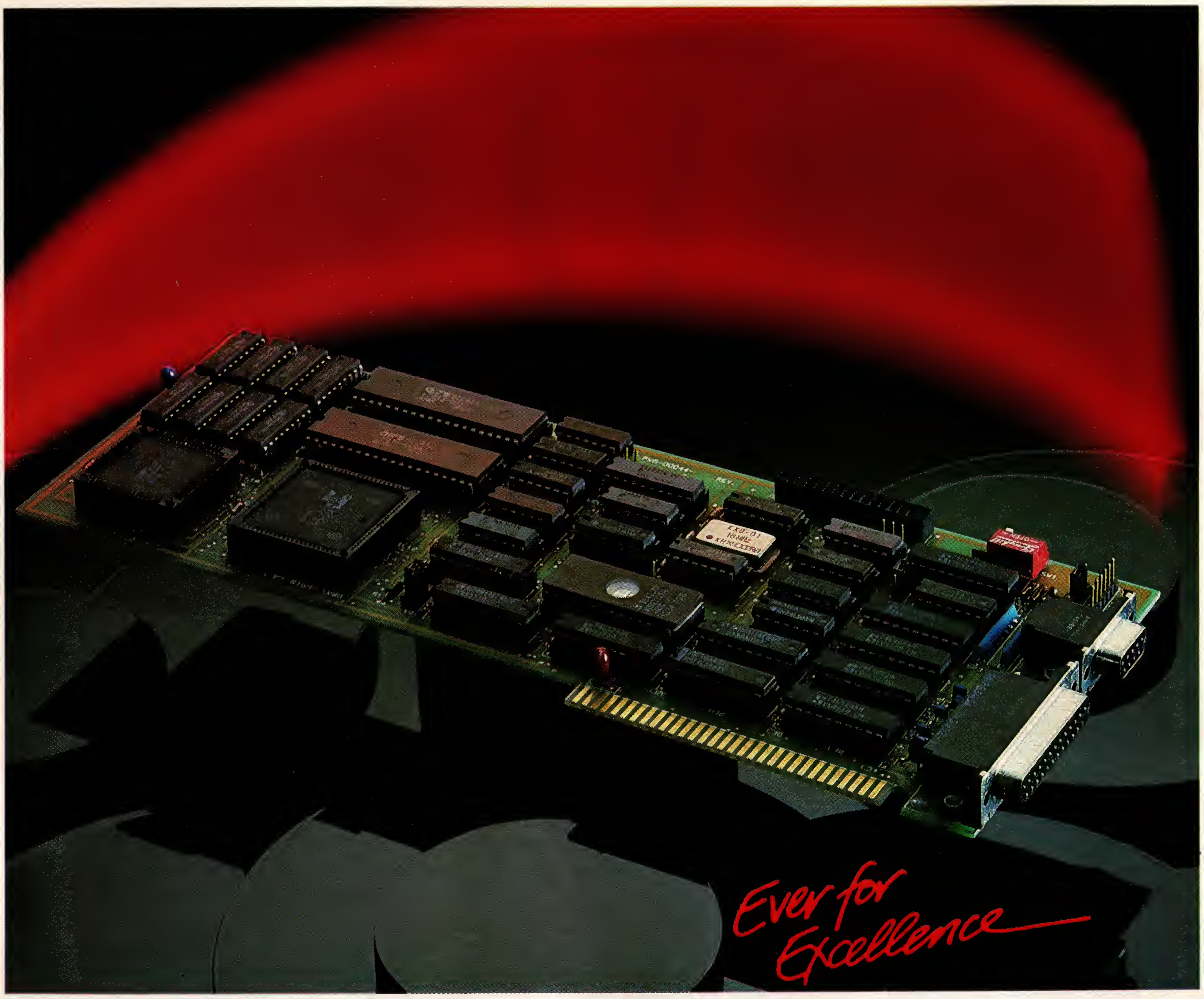
What is meant by a *smooth curve*? The intuitive idea of smooth curvature comes mostly from looking at what happens when flexible materials are bent. In fact, the traditional tool for drawing smooth curves is a drafting instrument called a spline, a flexible strip that can be anchored at points on the paper.

The curvature of a spline is described by a series of cubic equations. At the point where the curve changes from one cubic equation to another (the *knot*), the slopes of the two cubics are equal; thus the curve is smooth through those points. As mentioned earlier, a cubic equation can be fitted to any four points; it can be fitted equally well to two points and two slopes. The values of the slopes are not known at the outset, but they can be assigned unique values by specifying that the second derivative—the rate of change of the slope—must be 0 at the knots.

The remaining question is what to do at the ends of the curve. The most common approach is the so-called natural spline condition, which stipulates that the second derivative should be zero at the endpoints as well as the knots. A more satisfying type of cubic spline, devised by Carl de Boor (see the references at the end of this article), specifies that the second point and the next-to-last point are not knots—although the curve must pass through them, it does not change from one cubic to another as it passes. The first and last cubics are thus constrained by three points and one slope, rather than two points and two slopes. This helps control the curve's behavior at the ends.

Thus far, the assumption has been made that y is a function of x (that for any x only one y exists). In graphical terms, this means that only one pixel in each column can be illuminated, and the curve cannot pass above or below itself. This seems a severe restriction, and it has an easy resolution. Instead of having y be a function of x, x and y can both be treated as functions of another variable called t. Then the curve can be drawn by calculating both x and y for given values of t. The curve then can be two dimensional—it can wander all over the screen instead of just varying in height as it goes from left to right.

Several methods can be used to obtain t. In general, each point's t value should be greater than that of the previous point. This will force the resulting curve to pass through the points in the order in which they were entered. It may be satisfactory to say that $t = 1$ for the first point entered, $t = 2$ for the second, and so on, but SMOOTH.PAS takes a more sophisticated approach. The program begins with $t = 0$ and adds to t, at each point, the straight-line distance between that point and the previous one. This makes t roughly proportional to the length of the curve, so that the curve-fitting algorithm "knows" whether any two consecutive points are nearby or far apart. Thus, the program can decide how much curvature is acceptable between points: if points are close together, the line joining them should be relatively straight; if they are far apart, a kinkier line is allowed.



Enhancer™... EGA card, $399

The EVEREX Enhancer is the IBM compatible EGA card that delivers value with price, performance, and features!

No Options Needed.

With other EGA cards you have to pay extra for options like 256K screen memory and a parallel printer port. With the Enhancer these features are standard and there are no extras to buy.

11 Display Modes

There's 16 colors from a palette of 64 in 640x350 resolution. Text is displayed in readable 8x14 character format in text mode. And the Enhancer's monochrome mode can support monochrome graphics on a monochrome monitor. In addition to 6 EGA modes and a monochrome mode, there's 4 color modes.

Software Switched

The Enhancer comes with EGMODE menu-driven software that lets you change display modes, even monitors, without opening your chassis, with the push of a button. Useful help windows take all the mystery out of each possible display mode.

The Perfect Adapter

For enhanced color graphics, monochrome text, and standard color text and graphics, the EVEREX Enhancer is the perfect adapter to pair up with your software library. Call your local dealer for a demo, or call EVEREX to get all the facts on our new line of video adapters.

1-800-821-0806 1-800-821-0807

In California

48431 Milmont Dr.
Fremont, CA 94538
(415) 498-1111

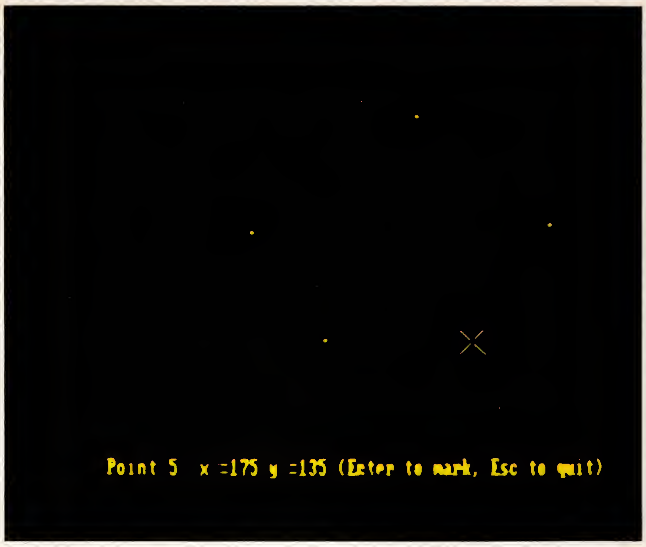
EVEREX

EVER for EXcellence

Enhancer is a trademark of EVEREX SYSTEMS, INC. IBM is a trademark of International Business Machines Corp.

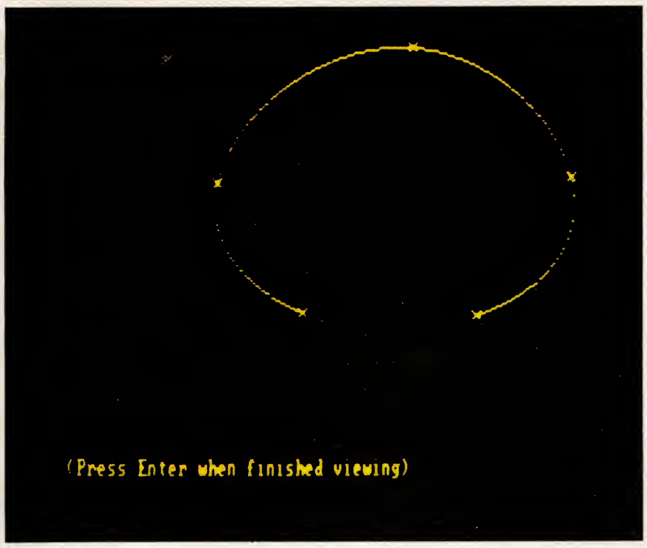
CIRCLE NO. 139 ON READER SERVICE CARD

PHOTO 1: *Selection of Points*



The program lets the user enter values for x and y by moving the cursor around the screen and marking points. At least four points are needed; the number of the next point to be entered is shown at the bottom of the screen.

PHOTO 2: *Resultant Curve*



When all the required points have been entered, the program calculates the cubic equations that are necessary to link them. The cubics are calculated by simultaneous equations with two known points and two known slopes.

THE DIVINING ALGORITHM

If x and y are treated as functions of t, then two spline curves must be computed, one for x and one for y. The computation of each spline curve is itself broken into two steps.

First, the coefficients of all the cubics that are chained together to make the complete curve are computed. The coefficients are found by solving simultaneous linear equations. At the knots, the values of t, t^2, t^3, and x (or y) are known, as are the second derivatives (which are all 0); in addition, the first derivatives (slopes) must match at the points where the segments join. Only the coefficients are unknown, and because none of the coefficients is raised to a power, the simultaneous equations are linear even though they describe cubic polynomials. The equations are solved by a process called *Gaussian elimination* (explained in the Sedgewick book listed herewith).

The second step is to compute x (or y) for the desired values of t. Because a different cubic is used between each pair of knots, the first step in handling a given t is to find out which knots it falls between. When this has been done, the appropriate set of coefficients can be looked up and the cubic can be evaluated.

The heart of the program is **spline**, a procedure designed to be used unchanged in other programs. The only external definitions it relies on are constants **arraysize** and **splinesize** and types **point_array** and **spline_array**.

The input to **spline** consists of two arrays, **f** and **t**, and an integer, **count**, which tells how many elements of **f** and **t** are used. For any **n** between 1 and **count** inclusive, **f**[**n**] and **t**[**n**] define a known point. The routine assumes that **f** is a function of **t**, that the arrays are sorted so that **t** is in ascending order, and that no value of **t** occurs twice.

The output consists of an array **calct** containing equally spaced values of **t**, and an array **calcf** containing the corresponding interpolated values of **f**.

The program decides how much curvature is acceptable between points: if points are close together the line joining them is relatively straight; if they are far apart a kinkier line is allowed.

The number of calculated points is specified by the constant **splinesize**. In this program, **calct** is never used because the final display is a plot of **x** versus **y**. **Calct** would be important if the interpolation were one-dimensional rather than two-dimensional.

The remainder of the program provides a quick and simple demonstration of **spline**. Note that because a major goal was to keep the program short, the routines other than **spline** are not particularly efficient. With the procedure **get_points**, the user inputs a set of points by moving the cursor around on the screen (see photo 1). **Get_points** treats both **x** and y as functions of **t**, which is the total linear distance between points. After the spline has been computed, the procedure **display** displays it on the screen (see photo 2). Several examples of the different curves that can be produced with this program are presented on the opening pages of this article, 112 to 113.

Spline will find purpose in many applications: especially attractive is its ability to store graphics images as only a few points, which is an economical use of memory. Perhaps what makes the program most useful is its ability to determine the distance between points and use this key in producing the smoothest of smooth curves.

REFERENCES

de Boor, Carl. *A Practical Guide to Splines*. New York, NY: Springer-Verlag, 1978.

Sedgewick, Robert. *Algorithms*. Reading, MA: Addison-Wesley, 1983.

Michael A. Covington, Ph.D., is a research associate at the Advanced Computational Methods Center, University of Georgia, where he is implementing a Prolog-like language on a CYBERPLUS supercomputer.

The assistance of McLowery Elrod, Ph.D., on this article is gratefully acknowledged.

LISTING 1: SMOOTH.PAS

```
program smooth_curve;

{ Turbo Pascal demonstration of parametric cubic splines }
{ Michael A. Covington                              1986 }

{ This program draws a smooth curve connecting points    }
{ marked on the screen by the user.  The points do not   }
{ have to define a function; internally, both X and Y    }
{ are functions of T, the approximate distance traveled  }
{ as the curve moves around the screen.                  }

{ NOTE: We are using 640x200 graphics but treating it as }
{ 320x200.  Thus all X coordinates are multiplied by 2   }
{ before being plotted on the screen.  The graphical     }
{ cursor occupies the odd-numbered columns in which data }
{ is never plotted.                                      }

const arraysize  = 100;  { maximum number of known points }
      splinesize = 400;  { number of points that are
                           calculated to draw the curve }

type  point_array  = array[1..arraysize] of real;
      spline_array = array[1..splinesize] of real;

var x,                   { x-coordinates of known points }
    y,                   { y-coordinates of known points }
    t:    point_array;   { t-coordinates of known points }

    count: integer;      { number of known points }

    calcx,               { x-coordinates of calculated points }
    calcy,               { y-coordinates of calculated points }
    calct: spline_array; { t-coordinates of calculated points }
      { CALCT is not actually used in this program, but is  }
      { included because other similar programs may use it. }

    reply: char;         { user's reply to 'Again?' }

procedure get_points;

   { Allows the user to move a cursor around the screen and }
   { mark points. X, Y, and T are recorded for each point.  }

   const  esc     = #27;
          enter   = ^M;
          up      = 'H';
          down    = 'P';
          left    = 'K';
          right   = 'M';
          beep    = ^G;

   var
     key: char;
     ix,             { current screen position }
     iy:  integer;

   function inkey: char;
     { Waits for a key to be pressed and returns its code.  }
     { Returns the second byte if a 2-byte key is pressed.  }
   var
     c: char;
   begin
     read(KBD,c);
     if keypressed then read(KBD,c);
     inkey := c
   end;

   procedure graphics_cursor(ix,iy,color: integer);
     { Puts the graphics cursor at ix, iy,      }
     { which are x and y expressed as integers. }
   var
     i,dy,dx: integer;
   begin
     for i:=1 to 5 do
       begin
         dy:=i;
         dx:=i+i+1;
         plot(ix+ix+dx,iy+dy,color);
         plot(ix+ix+dx,iy-dy,color);
         plot(ix+ix-dx,iy+dy,color);
         plot(ix+ix-dx,iy-dy,color)
       end
   end;

   function dist(x1,y1,x2,y2:real):real;
     { Distance between two points }
   begin
     dist := sqrt(sqr(x1-x2)+sqr(y1-y2))
   end;

   function same_point_again:boolean;
     { True if the current point is the same as }
     { the most recently entered point.         }
   begin
     if count < 1 then
        same_point_again := false
     else
        same_point_again :=
           (ix = x[count]) and (iy = y[count])
   end;

begin { get_points }

   textmode;
   clrscr;
   writeln('Move cursor with arrow keys.');
   writeln('Mark points with Enter key.');
   writeln('Press Esc when finished.');
   writeln;
   writeln('You must mark at least 4 points;');
   writeln('Esc will not work until you do.');
   writeln;
   writeln('You cannot mark more than ',arraysize,' points.');
   writeln('You cannot mark the same point twice unless');
   writeln('you have marked a different point in between.');
   writeln;
   writeln('Erroneous input will result in a beep.');
   writeln;
   writeln('Press Enter to begin...');
   readln;

   hires;
   hirescolor(yellow);
   { graphbackground(black); Uncomment to run on EGA }

   ix:=160;
   iy:=100;
   count:=0;
   key := ' ';

   while (count<4) or (key<>esc) do
     begin
       gotoxy(1,25);
       write('Point ',count+1,'  x =',ix,' y =',iy,
                      ' (Enter to mark, Esc to quit)   ');

       graphics_cursor(ix,iy,1);
       key := inkey;
       graphics_cursor(ix,iy,0); { erase previous cursor }

       case key of
         up:    iy:=iy-1;
         down:  iy:=iy+1;
         left:  ix:=ix-1;
         right: ix:=ix+1;
         enter: if same_point_again or (count=arraysize) then
                    write(beep)
                else
                  begin
                    count := count+1;
                    x[count] := ix;
                    y[count] := iy;
                    if count=1 then
                       t[count] := 0
                    else
                       t[count] := t[count-1] +
                            dist(x[count],y[count],
                                 x[count-1],y[count-1]);
```

Call for latest prices.

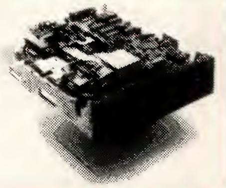
FREE SHIPPING

in the Continental United States via UPS Ground.

NO SURCHARGE FOR MasterCard OR VISA

30 DAY Money-Back Guarantee

Floppy Disk Drive



Also available for AT™ in Gray Color.

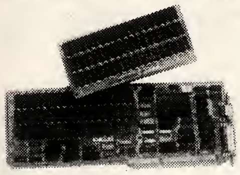
MITSUBISHI ELECTRONICS

$95

Half-Height, DS/DD

PC's Limited AT Multifunction Card

- Expandable To 3 Meg (1.5 on Board/1.5 on Piggy Back Board)
- Supports 64 or 256K Rams
- Parallel Port
- Serial Port (2nd Serial Optional)



$199 w/0K

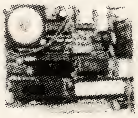
Piggy Back Board $59 w/0K

PC's Limited Mini I/O

- Serial Port
- Parallel Port
- Clock
- Software
- Fits In Short Slot

$99

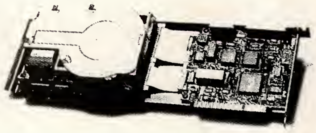
300/1200 Baud Hayes Compatible Modem Fits in Short Slot



$159

PC'S LIMITED 20 MEG HARD DISK ON A CARD

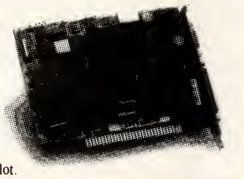
$409



- 3½" Winchester Hard Disk
- Easy to install
- One Year Warranty

UNIVERSAL GRAPHICS CARD II

$199

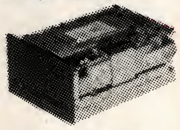


- Fully compatible with IBM CGA, Plantronics COLORPLUS, IBM Monochrome, and Hercules compatible graphics modes.
- Includes Parallel Port and fits in a "short" slot.

Seagate

20 30 and 40 MEG High Speed 40 MS Access Time Hard Drives for AT™

Uses Linear Voice Coil Activator.



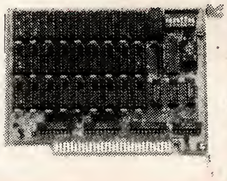
Heads park automatically at power down.

20 MEG $579
30 MEG $699
40 MEG $819

Includes Seagate Full Height Hard Disk, Cable, and Mounting Rails. Boots from Hard Disk. One Year Warranty.

PC's Limited PC-576 RAM Board

- Expandable to 576K
- Supports 64K or 256K RAMS
- Fits in Short Slot



$69 W/OK

PC'S LIMITED Six Function Card

- Upgradable to 384K
- Clock/Calendar
- Includes Software
- Parallel Port
- Serial Port
- Game Port

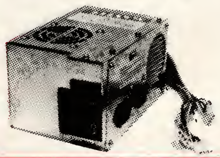
Two Year Warranty

w/0K $99 w/384K $149

SOLVE YOUR POWER PROBLEM.

XT™ POWER 130W

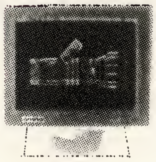
$99



Directly replaces power supply in PC.™ Fully XT™ compatible. One Year Warranty

MONO-II

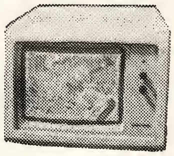
$129



- Flat screen
- Tilt and swivel base
- New circuit yields outstanding display

PC'S LIMITED RGB-1 MONITOR

$459

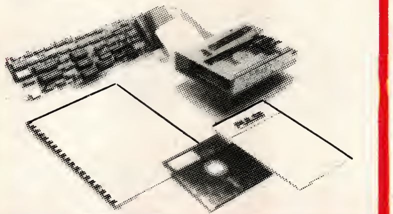


- .31MM dot pitch
- RGB input
- 12-inch screen
- 690x240 resolution
- 16 colors

Wangtek 60 MEG Tape Backup System

- Backs up 5 Megs per minute
- Runs under MS DOS or PC DOS, Novell, Xenix
- Pulse Software Included

$795



Tandon HD ▫ Western Digital Controller

20 MEG Hard Disk System for PC™

Internal $409

Includes Tandon Hard Disk, Western Digital Controller, Cables, Manual, Software, and Mounting Hardware. Boots From Hard Disk One Year Warranty

Or add a second 20 MEG Hard Drive to your system with the PC's Limited Hard Disk Kit—Two 20 MEG Hard Disks daisychained together, plus a controller card—**$695.**

Our Hard Disk Systems are compatible with the latest versions of the following Computers: IBM PC, IBM XT, AT&T 6300, Compaq Deskpro, Tandy 1000, Tandy 1200, PC's Limited Turbo PC, Leading Edge PC (Both Models), Sperry PC, Wyse PC, Televideo PC, Faraday Mother Boards, Corona PC, Eagle PC, ITT PC, and most other Compatibles.

PLEASE SPECIFY YOUR COMPUTER TYPE WHEN ORDERING.

WARRANTY

For Warranty Service: One year Limited Warranty on PC's Limited products. Contact Technical Support for a Return Authorization Number (RMA). Returns must be accompanied by your RMA, the invoice, and a brief explanation. During the Warranty Period, PC's Limited will repair or replace items at our option.

30-DAY TOTAL SATISFACTION GUARANTEE

Any item bought from PC's Limited may be returned within 30 days from the date it was shipped for a full refund of your purchase price. Returned items must be as-new, not modified or damaged, with all warranty cards, manuals, and packaging intact. Returned items must be shipped prepaid and insured, and must bear a PC's Limited Credit Return Authorization (CRA) on the shipping label. Call PC's Limited's Customer Support Department for CRA. No credits issued after 30 days from date of shipment.

CONSUMER TIP

When shopping for PC products, ask our competitors about *their* refund policies.

PC'S LIMITED™

SALES CALLS OUTSIDE TEXAS, 1-800-426-5150
SALES CALLS INSIDE TEXAS, 1-800-252-8336
1611 Headway Circle, Building 3, Austin, Texas 78754
Sales Calls from anywhere in the country, (512) 339-6962
Technical Support Calls, (512) 339-6963
Customer Service Calls, (512) 339-6964 and MCI MAIL: PC's LIMITED
Telex No. 9103808386 PC LTD FAX (512) 339-6721

Panasonic, TEAC, Mitsubishi, Intel, IBM, Hercules, Plantronics, Seagate, Western Digital, AT&T, Compaq, Tandy, Leading Edge, Sperry, Wyse, Televideo, Faraday, Corona, Eagle, and ITT are registered trademarks of their respective companies. IBM PC, IBM XT, IBM AT, are trademarks of IBM Corporation. Leading Edge PC, Sperry PC, Wyse PC, Compaq Deskpro, Televideo PC, Corona PC, Eagle PC, AT&T 6300, Tandy 1000 and Tandy 1200 are trademarks of their respective companies.

Some quantities may be limited. PC's Limited sells equipment certified to be compliant with FCC Class B standards. All prices are subject to change without notice.

Ad Number A08/86

CIRCLE NO. 145 ON READER SERVICE CARD

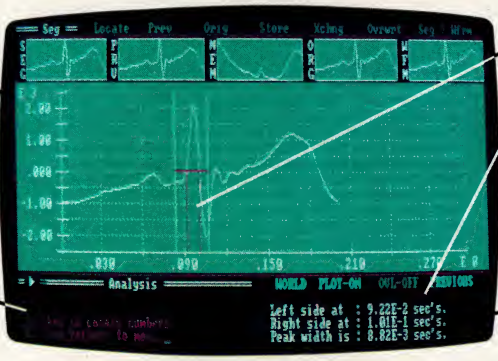
```
                  plot(2*ix,iy,1)
                end
        else  { do nothing };
      end { case }
    end
end;

procedure spline(f,t:point_array;
                 count:integer;
                 var calcf,calct:spline_array);

{ Fits a cubic spline to F[1..COUNT] as a function of  }
{ T[1..COUNT], then calculates equally spaced values   }
{ of T, places them in CALCT[1..SPLINESIZE], computes  }
{ the corresponding interpolated values of F, and      }
{ returns them in CALCF[1..SPLINESIZE].                }

{ At least 4 points must be given. The points are }
{ assumed to be sorted in order of increasing T.  }

{ Adapted from the FORTRAN routine CUBSPL (Carl de Boor,   }
{ A PRACTICAL GUIDE TO SPLINES, New York: Springer, 1978). }

var
   a,b,c: point_array;  { spline coefficients }
   d,e,g,h,             { used in finding spline coefs. }
   wt,dt:    real;      { used in calculating points to plot }
   i,j:      integer;   { loop counters }

begin

   { Compute first differences of T and F }

   for i:=2 to count do
     begin
       b[i] := t[i]-t[i-1];
       c[i] := (f[i]-f[i-1])/b[i]
     end;

   { Take care of beginning of curve }

   c[1] := b[3];
   b[1] := b[2] + b[3];
   a[1] := ((b[2] + 2*b[1])*c[2]*b[3] + b[2]*b[2]*c[3]) / b[1];

   { Forward pass of Gaussian elimination }

   for i:=2 to count-1 do
     begin
       g   := -b[i+1] / c[i-1];
       a[i] := g*a[i-1] + 3*(b[i]*c[i+1] + b[i+1]*c[i]);
       c[i] := g*b[i-1] + 2*(b[i]+b[i+1])
     end;

   { Take care of end of curve }

   g := b[count-1] + b[count];
   a[count] := ((b[count]+g+g) * c[count] * b[count-1]
               + b[count] * b[count]
               * (f[count-1]-f[count-2]) / b[count-1]) / g;
   g := -g / c[count-1];
   c[count] := b[count-1];

   { Complete the forward pass }

   c[count] := g*b[count-1] + c[count];
   a[count] := (g*a[count-1] + a[count])/c[count];

   { Back substitution }

   for i := count-1 downto 1 do
      a[i] := (a[i]-b[i]*a[i+1]) / c[i];

   { Generate cubic coefficients }

   for i:=2 to count do
     begin
       d := (f[i]-f[i-1])/b[i];
       e := a[i-1] + a[i] - 2*d;
       b[i-1] := 2*(d-a[i-1]-e)/b[i];
       c[i-1] := (e/b[i])*(6/b[i])
     end;

   { Compute the points to be plotted as function of WT }

   wt := 0;
   dt := t[count] / (splinesize-1);
   j  := 1;

   for i:=1 to splinesize do
     begin
       { Ensure that wt is between t[j] and t[j+1] }
          while t[j+1] < wt do j:=j+1;
       { Calculate a point }
          calct[i] := wt;
          h := wt - t[j];
          calcf[i] := f[j]+h*(a[j]+h*(b[j]+h*c[j]/3)/2);
       { Move to next point }
          wt := wt+dt
     end
end;


procedure fit_curve;
begin
  gotoxy(60,25);
  write('Calculating...');
  spline(x,t,count,calcx,calct);
  spline(y,t,count,calcy,calct)
end;


procedure display;

{ Displays the fitted curve and the original points. }

var i,k,ix,iy: integer;

begin

  hires; hirescolor(yellow);
  { graphbackground(black); Uncomment to run on an EGA }

  { Draw crosses at the original points }
     for i:=1 to count do
       begin
         ix := 2*round(x[i]);
         iy := round(y[i]);
         draw(ix-4,iy-2,ix+4,iy+2,1);
         draw(ix+4,iy-2,ix-4,iy+2,1)
       end;

  { Plot the spline curve }
     for i:=2 to splinesize do
        draw(round(2*calcx[i-1]),
             round(calcy[i-1]),
             round(2*calcx[i]),
             round(calcy[i]),
             1);

  gotoxy(1,25);
  write('  (Press Enter when finished viewing)   ');
  readln
end;


{ main program }

begin
    repeat

      get_points;
      fit_curve;
      display;

      textmode;
      write('Again? (Y/N)  ');
      readln(reply)
      until reply in ['N','n']
  end.
```

New from Macmillan Software

ASYSTANT scientific software. Flexible menu-driven environment gets you up and running–fast.

Fully integrated analysis, graphics, and processing functions operate directly on large datasets.

Lets you interact visually with your data. Just scroll, select curve segments, and watch your results appear.

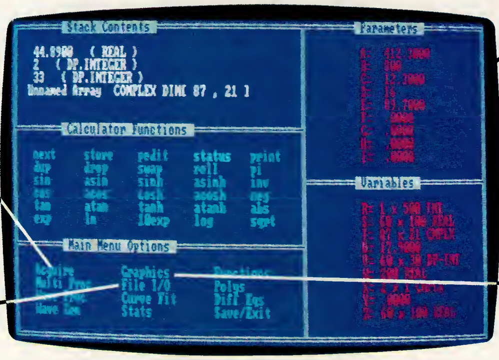


Unique features give you flexibility not found in most menu-driven packages.

On-line help is always available by typing "?".

Acquired data is immediately available for analysis and graphics. You never have to shuffle programs.

Gives you more power for the price than any other software for the IBM PC.



A single keystroke lets you import data from spreadsheet and other data file formats.

Presentation-quality scientific graphics easily output to plotters.

The PC software of choice when you don't want to program.

Now get the power of widely acclaimed ASYST™ Scientific Software in a stand-alone menu-driven package: ASYSTANT Ready-to-Run Scientific Software.™ Easy to learn and use whatever your computer background, ASYSTANT is a fully integrated analysis and graphics package. It's versatile, too. Unlike other menu-driven products, ASYSTANT gives you an extensive feature set and macro capability to tackle your toughest applications.

Built-in functions include FFT, smoothing, integration, differentiation, curve fitting, statistics, differential equations, and matrix and polynomial operations. And all analysis functions are fully linked to powerful graphics capabilities, including axonometric and contour plotting. ASYSTANT menu-driven scientific software–just **$495.**

Optional data acquisition with ASYSTANT +. Get all the features of the ASYSTANT package, *plus* one-touch data acquisition, with ASYSTANT +. No extensive prompt lists. No tedious set-up. Just choose the "metaphor" of the collection instrument you wish to simulate–such as XY or strip chart recorder, data logger, or signal averager–and you're ready to go. Only **$895.**

Free technical support. 60 days of free technical support help you design, set up, and fine-tune the perfect system. It's your guarantee of trouble-free operation.

30-Day No-Risk Offer.
CALL 1-800-348-0033
In New York state, (212) 702-3241.

Macmillan Software Co.
An affiliate of Macmillan Publishing Company
630 Third Avenue, New York, NY 10017

RAMpage AT™-The Cure for

the AT Blues.

Doc plays sax down at Al's on Wednesday nights. That's blues night at Al's. And sometimes, if you shut your eyes and listen to that sultry, wailing music, you'd swear there's more than just one, lone sax playing. Doc knows his sax...been playing the same one over thirty years. And he plays it for all it's worth.

At AST we know the IBM® PC AT. We were the first to produce and ship an AT enhancement product. Like Doc's sax, your PC AT is a remarkable instrument with great potential. But if you're not using our RAMpage AT expanded memory solution, you're not playing it for all it's worth.

THE AST PREMIUM SERIES™

A Sure Cure For The AT Blues. Our RAMpage AT with up to 2 megabytes of RAM and DESQview™ multitasking/windowing software lets your PC AT soar to its maximum potential. And lets you realize your full AT investment value.

RAMpage AT For Multitasking Power. The RAMpage AT's powerful features combined with the PC AT's speed offer fast and functional multitasking capability.

Only with RAMpage AT can you sort a database, watch on-line stock quotations, calculate a spreadsheet, print labels and write a letter using your favorite word processing package...all at the same time...on the same PC AT. And with DESQview's windows you'll view and access up to nine applications instantly. You'll work faster and avoid time-consuming workflow interruptions.

Compatibility You Can Rely On. Our Enhanced Expanded Memory Specification is simply a more powerful, fully compatible version of the Lotus®/Intel®/Microsoft® (LIM) Expanded Memory Specification (EMS). With RAMpage AT you can use *All EMS Software.*

LOTUS COMPATIBLE · Lotus · ASK YOUR DEALER FOR APPROPRIATE CONFIGURATIONS

It's even certified compatible by Lotus Development for 1-2-3® Release 2 and Symphony® 1.1.

Powerful Features. RAMpage AT offers up to two megabytes of expanded memory that can also be used as extended memory, DESQview multitasking/windowing software, SuperPak™ utilities and our two-year warranty. Additionally, AST offers several models of the RAMpage AT-Pak, an optional module featuring serial, parallel and game ports.

Play Your AT For All It's Worth With RAMpage AT. For more information call our Product Information Center **(714) 863-1480.** Or send the coupon to AST Research, Inc., 2121 Alton Avenue, Irvine, California 92714-4992.

RAMpage AT and SuperPak trademarks of AST Research, Inc. IBM registered trademark of International Business Machines Corp. DESQview trademark of Quarterdeck Office Systems. Lotus, 1-2-3 and Symphony registered trademarks of Lotus Development Corp. Microsoft registered trademark of Microsoft Corp. Intel registered trademark of Intel Corp. The RAMpage AT Expanded Memory Board has successfully passed compatibility testing with 1-2-3 Release 2 and with Symphony 1.1 on an IBM AT with PC-DOS 3.1. Two megabytes of memory were installed. The Lotus logo is used by permission of Lotus Development Corp.

AST RESEARCH INC.

CIRCLE NO. 140 ON READER SERVICE CARD

Yes, I want the cure for the AT blues. Please send me more information about RAMpage AT.

Name ____________________

Title ____________________

Company ____________________

Address ____________________

City ____________ State ______ Zip ______

( )

Phone ____________________

AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714-4992.

PCTR/AT-B1


DATAFLEX®
The World's Best Selling Multi-User Database For Five Years Running.
DATAFLEX

Software Sprites

Although the IBM PC is hindered by a lack of animation hardware, some unusual programming work-arounds can yield rather sophisticated results.

MICHAEL ABRASH and DAN ILLOWSKY

IBM PC animation is crude in comparison to arcade-quality games. Where game machines—which include personal computers far less expensive than the PC—routinely animate many large objects without flicker or interference, PC animation tends to feature a few small, flickering images. Worse, odd color and fringe effects occur when objects overlap on the PC, instantly destroying the illusion by revealing the animated objects as nothing more than arrangements of pixels. This is the case because the PC simply was not built as a game machine, and consequently lacks basic animation hardware. It is often true, however, that clever software can replace hardware. This holds for animation: software emulation of the hardware animation capabilities of game machines is a realizable, albeit difficult, goal for the PC.

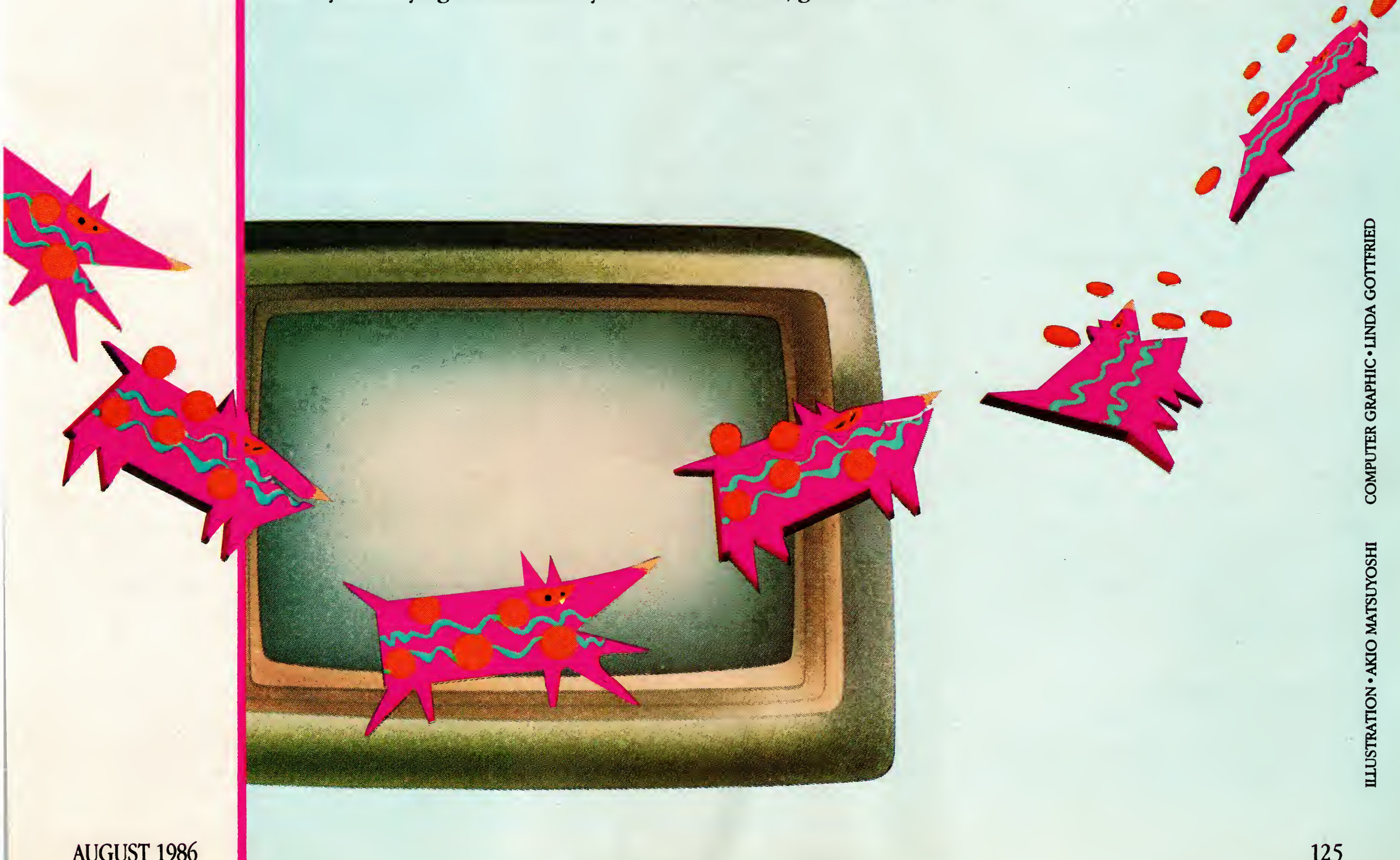


ILLUSTRATION • AKIO MATSUYOSHI   COMPUTER GRAPHIC • LINDA GOTTFRIED

The key to sophisticated animation for the PC is the emulation by software of the hardware sprites that make animation on game machines so spectacular, yet simple to implement. From a programmer's perspective, a hardware *sprite* is a screen object for which all the details of drawing the corresponding image are handled in hardware. Animation hardware generally supports many sprites, allowing multiple objects to be animated simultaneously. When sprites overlap, the sprite with the higher priority is displayed in front of the others, making the objects seem solid and the screen three-dimensional. Objects with lower priorities are clearly visible around the edges and through transparent areas of higher-priority sprites. Sprites are nondestructive, so the programmer never has to worry about restoring the background or other sprites when one is moved. Once a sprite has been reprogrammed with the new location of an object, the hardware assumes all responsibility for updating the screen correctly, leaving the programmer free to concentrate on the design of a program rather than the messy details of animation.

The IBM PC has no sprite hardware, nor anything resembling it. Animation for the PC must be performed entirely by the 8088, a task of such complexity and realtime constraints as to make software sprites seem impossible. The history of microcomputers, however, has proven many times over that the seemingly impossible often is achievable if a programmer will not let such an assumption get in the way of searching for a solution.

This article traces the design and implementation of a software sprite driver that concedes nothing save a little time and memory to hardware sprites. The actual implementation also illustrates many features of animation programming. Software sprites were created for both the IBM Color Graphics Adapter (CGA) and the Enhanced Graphics Adapter (EGA) and will take advantage of the little-known vertical interrupt feature of the EGA to make that implementation more powerful and program-independent. Finally, some finer points of realtime animation and a few design approaches through which animation programs using software sprites can achieve even more striking effects are discussed.

A SPRITE ENGINE

The animation driver must handle all the details of smoothly drawing multiple objects so that when they overlap, each will appear to exist in a separate plane. Software sprites will support object overlap, background preservation, display priority, transparency, and independent background processing—all the properties that make hardware sprites so visually effective—and one other feature as well: independent background processing.

The hardware in game machines draws sprites independently of the application (except when the program is reprogramming a sprite). The same sort of animation independence is desirable in software sprites. The EGA version of this driver is controllable from any program via a standard interface; it operates in the background, transparently and asynchronously to the calling program and apart from the interface calls. When an application program notifies the sprite driver of a change in the coordinates of an object, the driver automatically erases the object at its old location and redraws it at its new location during the next available drawing time. This driver is, therefore, appropriately called a sprite *engine* because once programmed it supports self-contained and complete animation. The CGA version requires the application to emulate the vertical interrupt feature of the EGA, which makes it less application-independent, but in all other respects it supports true software sprites.

To display only the end product of erasing all the objects and redrawing them (and thus avoid flicker), the cycle must be completed during nondisplay time.

Some familiarity with animation, including both the relocating of images to create the illusion of motion and the organization of display memory in 320-by-200, four-color graphics mode, is assumed on the reader's part. (See "Animation Techniques," Michael Abrash and Dan Illowsky, July 1986, p. 46 and "The IBM Color Graphics Adapter," Thomas V. Hoffmann, July 1983, p. 26.)

The first requirement in creating the sprite engine is an animation driver that causes each object to appear to be in a separate plane, so that objects clearly will pass behind and in front of each other. When an object is moved, the area it covered at the old location must be restored to the object or objects that are "behind" it, or to the background if no other objects were at that place. In addition, when the object is redrawn at its new location, it must be overlapped by more forward objects and must overlap more rearward objects. While other workable solutions to the overlap problem are available, the method outlined below keeps this driver simple and flexible.

A straightforward way to ensure that a certain object always appears in front of some objects and behind others is to redraw all the objects from scratch in the same order each time any object is drawn (in increasing order of priority—the object that appears to be nearest the background first, one that appears to be nearest the viewer last). A considerable amount of processing time is required here, the more so because all the objects must be erased (that is, restored to the background) before they can be redrawn. However, this process ensures that the higher-priority objects will be drawn over the lower-priority objects whenever the two overlap, which creates the illusion that the higher-priority objects are in front.

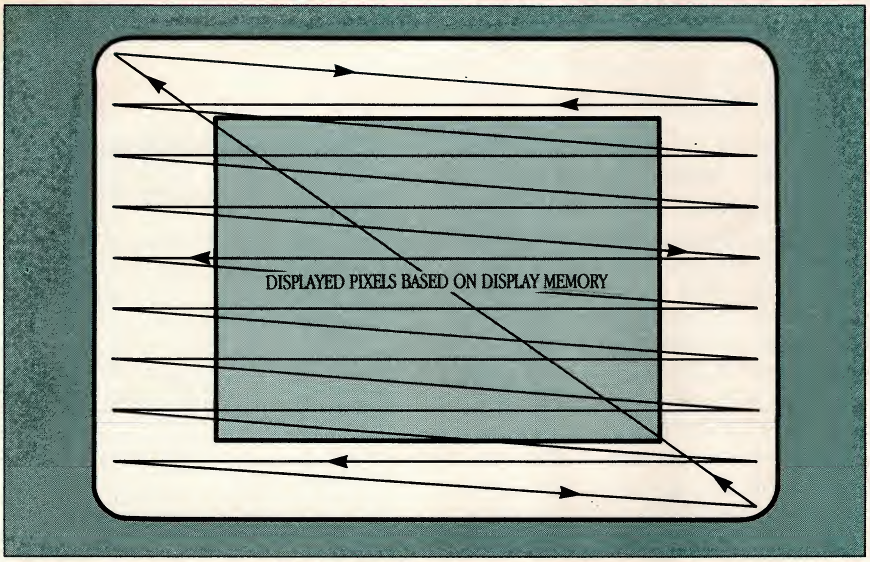
While this approach is fundamentally correct, problems remain. The objects will vanish frequently (and flicker as a result) as they are erased and redrawn. A special driver is required to allow rearward objects to show through and around the edges of the forward objects. In order to display only the end product of the cycle of erasing all the objects and then redrawing them (and thus avoid flicker), the entire cycle must be completed during nondisplay time. This explanation requires some familiarity with the nature of bit-mapped graphics and the display controller.

BIT-MAPPED GRAPHICS

The EGA and CGA are bit-mapped graphics adapters, which consist of two parts: memory and a display controller that constantly scans that memory, translates the contents of the memory into video data, and sends the video data to a monitor. The monitor then uses the video data to control the electron beam that is scanning pixels onto the screen. The correspondence of display memory to pixels is shown in figure 2 of "Animation Techniques" (July 1986, p. 49). However, this figure conveys only a static sense of the dynamic process of mapping display memory to pixels.

Pixels actually are drawn one at a time by the scanning electron beam,

FIGURE 1: *Electron Beam Scan*



The beam is turned off while moving through retrace upward or rightward. The portion of the beam that falls outside of display memory is called *overscan*.

which traces a pattern like that shown in figure 1 (of this article). This figure is a simplification of display monitor scanning, but illustrates two important points: the electron beam scans left to right while progressing downward, and margins are present at all four sides of the screen where the pixels generated by the display controller are not based on the contents of display memory.

The electron beam scans pixels onto the screen from left to right, then returns rapidly to the left margin upon receiving the horizontal synchronization pulse from the display controller. The video data sent to the display is based on the contents of display memory except during the left and right margins (the horizontal retrace period). Likewise, the beam progresses downward as it scans left to right, returning rapidly to the top of the screen upon receiving the vertical synchronization pulse. The vertical retrace period is that time at the top and bottom margins of the screen when display memory does not affect what is being displayed on the screen. The time required to scan the entire display and return to the upper left corner is called the *frame time*, while the time required to scan a single line and return to the left side of the screen is called the *scan line time*.

Together these two figures (figure 2 from the July article and figure 1 here) illustrate the direct correspondence of memory scanning to display scanning over time. At any one time, one area of memory (at most) is being scanned to provide video data for one area of the display. During the retrace periods, however, memory is not being scanned for video data at all, and bit-mapped video data is not being sent to the screen. As a result, any changes made to a given area of display memory during the retrace period will not be seen immediately; in fact, such a change will not be apparent until the point in the next frame when that area of memory is next scanned for video data.

Consequently, if a display memory byte is changed twice during the same retrace period, only the second change will show up on the screen. This is precisely the technique needed to rid the sprite engine of flicker; if objects are erased and redrawn during the same retrace period, the erasure will never be seen. To the viewer, the objects will appear to have moved from one location to the next with no transition.

Erasing and redrawing during a single retrace period, however, is more easily said than done. Even the highly efficient driver developed here requires a substantial amount of time to erase and redraw an object. A medium-size object, for example, takes roughly one or two milliseconds (ms) to draw (including erasing and redrawing), and *all* objects must be erased and redrawn every time any one object is moved (although the erase step may be skipped for certain objects, as discussed below). The next step, then, is to find adequate consecutive retrace time.

The horizontal retrace period clearly is too brief to be useful in supporting sprites. In 320-by-200 graphics

mode, for example, an entire horizontal scan line takes only 62.5 microseconds. Trying to do any redrawing during this period is out of the question . However, each full frame takes much longer—16.7 ms, a rate of 60 frames per second. Thus, the vertical retrace period provides sufficient consecutive retrace time to implement a useful sprite engine.

Note that the EGA's highest-resolution mode (350 scan lines) has virtually no vertical retrace period at all, so the sprite engine cannot be implemented. However, in the modes with 200 scan lines, including the CGA-compatible graphics modes and the 16-color graphics modes, just under one-quarter of each frame (about 4 ms) is in vertical retrace; this is ample time in which to perform effective animation. The CGA-compatible 320-by-200, 4-color mode is used in this article because it allows the same routines to be used in both the CGA and EGA versions and because handling multiple colors in a single image is much easier with the CGA mode than the EGA. However, the EGA's 320-by-200 and 640-by-200 16-color modes certainly could be supported by the sprite engine with modifications to the graphics driver.

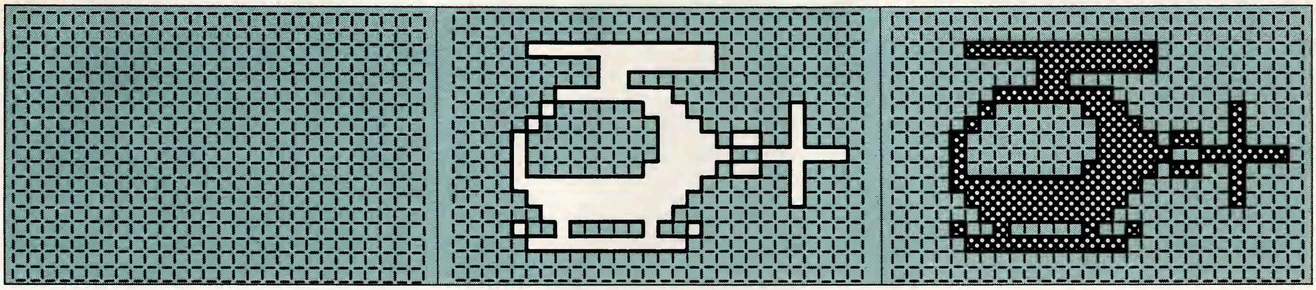
Thus, the problem of flickering objects is solved by performing all erasing and redrawing during the vertical retrace period in a 200-scan-line mode. Applications that demand higher vertical resolution require a different technique. As is often the case, no general solution exists: challenges must be dealt with on a case-by-case basis.

A SPRITE-LIKE DRIVER

The remaining problem is to draw objects in a sprite-like fashion, with one object clearly in front of the other and with no interference effects—and to do so reasonably quickly. The AND/OR driver solves this problem neatly.

Like other graphics drivers, the AND/OR driver modifies an area of display memory that controls a rectangular area of the screen; it operates from a table of bytes that describe the object to be drawn. The unusual aspect of the AND/OR driver is that each byte of the final image written into display memory is controlled by a pair of image table bytes. The first image table byte of each pair is used to mask off bits in display memory with the AND instruction; the purpose of the masking is to clear the bits controlling the pixels where the new object is to go without disturbing the bits controlling the pixels that the new object does not affect. The second image table byte of the pair is the actual

FIGURE 2: *AND/OR Driver Operation*



In the first frame, just the background is visible. In the second, the AND pass of the AND/OR driver clears those bits where the object will appear. Finally, the OR pass inserts the bits from the object itself without disturbing the background bits.

pixel data for the object, which is inserted into the masked display memory byte with the OR instruction.

This approach allows objects to cross without the odd color effects that are typical of exclusive-or drivers or the rectangular blanking fringe around objects that occurs with a byte-move driver, and allows the background to show through transparent (background-colored) areas of an object. (Any background-colored area may be made opaque simply by using the mask byte to mask out the background.)

Figure 2 illustrates how the two steps of the AND/OR driver work together to insert an object into display memory. The left frame shows the background before the object is drawn. The middle frame shows the display after the bits where the object is to appear have been masked off with the AND instruction (only those pixels where the object is to appear have been affected). The right frame shows the display after the object has been inserted into display memory with the OR instruction. All pixels other than those belonging to the object remain unchanged, so the object is inserted into display memory without interference effects.

Unfortunately, this approach is not particularly fast, even when in-line code is used, because two image table bytes must be read for every display memory byte modified. Also, because the AND and OR logical functions must be performed, little advantage can be derived from the use of the highly efficient string instructions. If faster drawing is required, a byte-move driver, which simply moves a rectangular block of image table data into display memory, can be used as a substitution for the AND/OR driver; in this case, however, it would cause the objects to be surrounded by blanking fringes, and transparency would not be supported.

Nevertheless, a driver like the byte-move driver is used for another purpose in the sprite engine: to erase the objects. An uncomplicated way to support an arbitrarily complex static background is to store the entire background in a separate buffer in normal memory that corresponds on a byte-to-byte basis with the display memory buffer, and then erase each object (restore it to the background) by copying the bytes corresponding to the area covered by that object from the background buffer to display memory. To do this, the driver simply calculates a single offset, which applies to both the background buffer and display memory, and then uses in-line code to copy the background data into display memory.

These two drivers put the last piece of the sprite engine into place. The screen will be updated each time any object moves: all objects will be erased by restoring their area of display memory to the background first and then redrawing them with the AND/OR driver, all during a single vertical retrace period. With this in mind, consider a unique feature of the EGA that makes the sprite engine efficient and easy to implement.

THE VERTICAL INTERRUPT

The graphics driver described here requires that the erasing and redrawing sequence start at the very beginning of

THE EGA'S VERTICAL INTERRUPT STATUS BIT

IBM suggests that each hardware interrupt handler check to make sure that the interrupt to which it is responding was generated by the device it handles, passing along the interrupt to the next handler for that interrupt if it is not. This approach, if followed by all devices, allows multiple devices to share the same interrupt level and expands the PC's device-handling capabilities in addition to reducing potential conflict between peripherals. The IBM EGA provides a vertical retrace status bit (bit 7) in the Input Status 0 register for just this purpose. IBM specifies that when an interrupt occurs on IRQ2, the routine handling the EGA's vertical interrupt should check this bit, passing the interrupt along to the next handler if the bit is low and handling it if the bit is high.

That, at any rate, is the theory. Two factors complicate the interpretation of the vertical retrace status bit, however, rendering it of limited usefulness, and, in fact, making it desirable in many circumstances to ignore the vertical retrace status bit entirely. First, some EGA-compatible adapters, including Quadram's QuadEGA+, return an inverted vertical retrace status bit, with the status low when the video adapter has generated the interrupt and high otherwise. This will be corrected in the near future, but by that time, tens of thousands of old-style boards will be in use. (To complicate the matter further, IBM's *Technical Reference Options and Adapters*, Volume 2, has the polarity of the vertical retrace status bit backward.) The developer is in danger here, as vertical interrupt-based software that runs perfectly on an EGA will lock up on some compatible adapters.

While methods are available to determine the polarity of the vertical retrace status bit via software for any given board, the need for this is much removed by the second complication, the physical nature of the vertical retrace status bit. The status bit is tied directly to the IRQ2 bus line, and as a result indicates only whether the IRQ2 line has been raised to generate an interrupt, *not* whether the interrupt was specifically generated by the EGA.

The upshot is that determining whether an IRQ2 interrupt was generated by an EGA vertical interrupt is not a simple matter of sampling a single bit. Considering that few peripherals use IRQ2, the best approach for a developer using the vertical interrupt might well be to forget about the vertical retrace status bit and assume that all IRQ2 interrupts are indeed vertical interrupts. This removes all danger of misinterpretation and speeds up interrupt response slightly.

If it is essential that other devices share IRQ2 with the EGA, a less-than-perfect solution is available. First, the polarity of the vertical retrace status bit must be determined. The easiest way to do this is to clear interrupts and toggle the vertical interrupt enable bit low and then high, forcing

the vertical retrace period, in order to maximize the amount of time available for updating the objects. Two problems arise on the CGA. First, the standard means of checking for the vertical retrace period on the CGA is to poll the vertical synchronization pulse status bit; however, because the pulse begins well into the period, a good portion of the retrace period is lost. (Some methods are available for finding the beginning of the vertical retrace period on the CGA, but they are neither entirely standard nor simple to explain.) Worse, the status bit must be polled constantly because the leading edge of the pulse must be found in order to allow maximum drawing time. This constant polling restricts the use of program time for other purposes. Although the polling could be interleaved into other program activities, or the program could be designed so it would take just under one frame time in between drawing times, approaches such as this make polling for synchronization the overriding factor in program design and thereby restrict the designer's options. While this approach was used in the CGA version of the sprite engine, the EGA provides a superior means of detecting the start of vertical retrace.

The EGA's *vertical interrupt* (discussed further in the sidebar above) simplifies retrace detection. When this feature is enabled, the EGA generates an interrupt on IRQ2 each time that the vertical display end setting is reached: this is the start of each vertical retrace period. (A 1982 edition of the IBM *Technical Reference* manual lists level 2 as the interrupt level of the CGA's vertical retrace, a feature that was never implemented.) The sprite engine must be

such an interrupt to occur. (Given that the vertical interrupt is in an enabled and cleared state, one is generated whenever its enable bit is toggled high to disable the vertical interrupt. Users should be aware of this when turning off the vertical interrupt and restoring the original IRQ2 interrupt vector.) Once the vertical interrupt has been forced, read the vertical interrupt status bit, then clear the interrupt and read the status bit again. The first state read is the vertical interrupt *active* status, the second is the vertical interrupt *not active* status. If the readings are identical, then another interrupt is being signaled on IRQ2; interrupts should be enabled and this other interrupt allowed to occur, then the entire process should be repeated until the vertical retrace status bit is made to toggle. When the status bit does toggle, an EOI should be issued to the interrupt controller before interrupts are enabled to wipe out the test vertical interrupt just generated because no handler was set up for it.

Once the polarity of the vertical retrace status bit is known, a vertical interrupt handler can be set up. The only information this handler can derive from the status bit is as follows: if, when handling an IRQ2 interrupt, the vertical interrupt is cleared and the vertical retrace status bit remains active, then another device is signaling an interrupt on the IRQ2 line. Unfortunately, this does not mean that the EGA is not signaling one as well, as two (or more) interrupts could be signaled simultaneously.

In the case where the vertical retrace status bit remains active when the vertical interrupt is cleared, one of two assumptions can be made: the handler can assume that *both* the vertical interrupt and another interrupt occurred, or the handler can assume that *only* another interrupt occurred. Too little information is available to make a full evaluation, and as a result this approach is not ideal. The handler can either deal with such cases by handling the interrupt as if it were a vertical interrupt and then passing it along to the next IRQ2 handler (assuming both interrupts occurred), or it can skip the vertical interrupt handling and go directly to the next IRQ2 handler (making the assumption that another interrupt occurred). These assumptions will not always be correct, but adding or missing an occasional vertical interrupt will have no noticeable impact on most animation applications, therefore this approach generally is an adequate solution to vertical interrupt handling.

Nevertheless, the simplest solution is to assume that IRQ2 is reserved for the EGA. Perhaps the next generation of the EGA will have a more useful status bit. Unfortunately, the hundreds of thousands of EGAs and compatibles already installed mean that the quirks of the vertical retrace status bit will be around for years.

—Michael Abrash and Dan Illowsky

set up as the interrupt handler for IRQ2, with the EGA vertical retrace interrupt enabled; the sprite handler will run virtually transparently in the background while the application program processes in the foreground.

To enable the vertical interrupt, bit 5 of the EGA's vertical synchronization end register (CRTC register 11H) must be set to 0. Each time the vertical interrupt occurs, it must be cleared by setting bit 4 of the register to 0, then reenabled by setting bit 5 to 0. Because bits 0 to 3 of this register define the character count at which the vertical synchronization pulse is to end, these bits must be set correctly for the current display mode whenever bit 4 or 5 is modified. Bit settings for all modes are provided in the IBM *Technical Reference, Options and Adapters,* volume 2, in the section on the EGA.

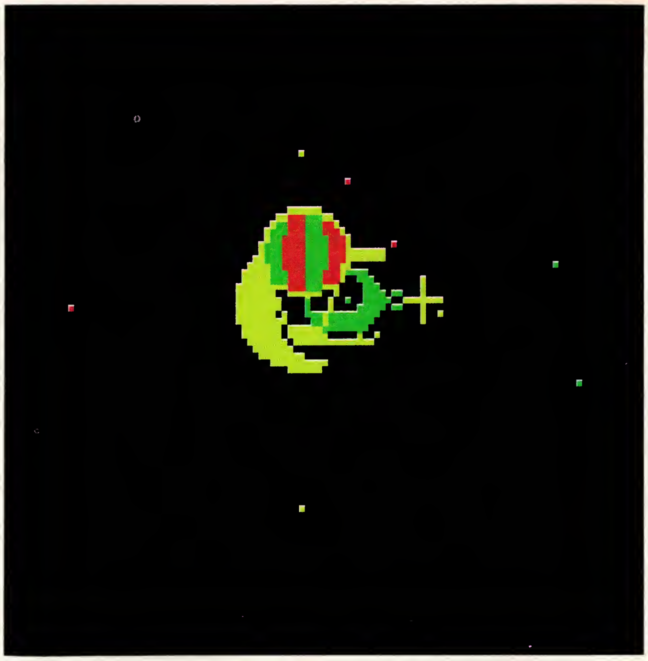
In order to use the vertical interrupt, the interrupt controller must be set to allow IRQ2 to occur by setting bit 2 of port 21H to 0. IRQ2 is handled by the routine pointed to by interrupt vector 0AH, at address 0000:0028, so this vector must be set to point to the vertical interrupt handler. However, devices other than the EGA may be using IRQ2 as well. In fact, in the PC/AT, the real-time clock interrupt and the hard-disk interrupt are among the eight interrupts cascaded through the IRQ2 interrupt. Consequently, any program using the vertical interrupt is mandated to restore the interrupt 0AH vector and the interrupt controller IRQ2 bit to their original states upon terminating.

The vertical interrupt handler also must make sure that each IRQ2 interrupt it handles was in fact generated by the EGA, and pass on to the routine

PHOTO 1: *Sprites in Action*

The balloon and helicopters are sprites; the moon, stars, and horizon are background. The helicopter windows are transparent to the background and lower-priority sprites. (Note the star visible through the red helicopter's window.)

PHOTO 2: *Overlapping Sprites*



In this enlarged view of the screen, the higher-priority balloon overlaps the green helicopter and obscures it. The red helicopter (out of view), were it to cross them, would obscure both because its priority is the highest of the three.

pointed to by the original interrupt 0AH vector any interrupts not belonging to the EGA. Bit 7 of the EGA's input status register 0 (port 3C2H) will indicate whether the vertical interrupt has occurred; if this bit is 1, the IRQ2 to which the vertical interrupt handler is responding was generated by the EGA.

This completes the design of the sprite engine, which will provide independent background processing, controlled by the application program through a standard interface. All drawing is done during the vertical retrace period, so no flicker occurs. Sprite-like animation is supported, with distinct precedence of objects; the background is preserved, and object transparency is supported. But can this driver handle enough objects to make it useful? The answer is yes, with a few tricks.

FUNCTIONAL PARTS

The program in listings 1, 2, and 3 is a fully functional application of the sprite engine. To see it in action on an EGA, assemble the listings with the Microsoft Macro Assembler (4.0 was used for testing), link them, and run the resulting .EXE file. To run the CGA version (which operates on an EGA but without the vertical interrupt), change the value equate for the **ega** flag in listings 1 and 2 to 0, then assemble, link, and execute as for the EGA version.

This program animates two helicopters (14 pixels high by 24 wide) and a balloon (20 by 16) against a backdrop of stars, a crescent moon, and a horizon (see photo 1). The helicopters have large windows, through which the stars, moon, balloon, and even the rearward helicopter can be seen (see photo 2). All other objects are visible around even the most detailed parts of each object. One helicopter periodically vanishes for a short time. No flicker is present when operating on the EGA, and the background is perfectly preserved. Moreover, the program spends most of its time in a wait loop that regulates execution speed, which leaves plenty of spare time for more sophisticated processing on the part of the program. It could, for example, detect collisions, keep score, or orchestrate the action on the screen. Because the vertical interrupt lets the sprite engine operate only when needed and without intervention from the main program, the main program can use all of display time productively if it needs to.

Listing 1 is the application program using the sprite engine (listing 2) and a graphics driver (listing 3). The EGA implementation differs markedly from the CGA version because the EGA is driven by the vertical interrupt. Only the EGA version is discussed here. Listings 1 and 2 contain some code delineated by **if...endif** conditional assembly blocks, with assembly conditional on the state of the **ega** or **cga** flag. When the **ega** flag is nonzero, the code appropriate for the EGA is assembled by the blocks starting with **if ega**.

The sprite engine breaks down into four functional parts: initialization, update, termination, and interrupt handler. These routines center around a queue containing all relevant information for the objects to be drawn, including a screen position and a pointer to an image table for each object. The queue may be predefined to support any number of objects by changing the **number_of_objects** variable. The object with the highest queue number has the highest drawing priority (it is drawn after all other objects) and so appears to be farthest forward. Any queue entry can be inactivated easily, so it is not required that all queue entries be active at once. This provides a simple mechanism whereby the relative display priorities of the objects being animated can be changed (see below).

The initialization, update, and termination sections can be called as near procedures when linked with an applications program; the interrupt handler updates the objects, as needed, on each vertical interrupt. The interface to the first three routines is described in the comments of listing 2.

The initialization routine, **initialize**, sets the sprite engine's internal variables to indicate that the object queue is empty. The address of the buffer containing the background information used to erase objects is set as passed from the application program, as well. In addition, **initialize** sets variable **need_to_draw_something_flag** to indicate that the screen need not be updated on the next vertical interrupt.

Next, **initialize** sets up the vector for interrupt 0AH (the vector used by IRQ2), to point to the routine **put_objects_on_screen**, which updates the screen as required on each EGA vertical interrupt. (The CGA version pushes the flags and executes a long call to **put_objects_on_screen** to emulate IRQ2.) The original interrupt 0AH vector is set aside for two purposes: to be restored at the end of the program and to allow IRQ2 interrupts not generated by the EGA to be passed on by the interrupt handler.

The state of IRQ2 is read from the interrupt controller at I/O address 21H and stored so it can be restored when the program ends. IRQ2 is then enabled, so that the vertical interrupt will be passed along by the interrupt controller to interrupt the CPU. Interrupts must be disabled while the state of the interrupt vector is modified, otherwise an interrupt might occur in the middle of modification and find an invalid (because it is partially changed) vector.

The update routine, **object_services**, is called by the application to register changes in the status of an object, such as a change of location or a new form table for internal animation. Internal animation of a sprite may be accomplished by defining several form tables for a single sprite, each containing one *frame* of the animated figure's motion. The software cycles through the sequence of form tables by repeated calls to **object_services**. (This is how the blades and tail rotor of the helicopter sprites are animated.)

The queue entry for the object with the number indicated by the application is updated on the basis of the information passed to **object_services**. Parameters such as the offset from the end of one scan line of the object to the start of the next and the offsets to call in the erasing and redrawing in-line code are calculated now, so that when the vertical interrupt occurs, these values can be read directly from the queue entry, saving precious time during the vertical retrace period. Finally, **object_services** sets **need_to_draw_something_flag** to true, to indicate that

the screen must be updated on the next vertical interrupt because the object's status has changed. As with **initialize**, interrupts are disabled while the queue is updated, preventing the vertical interrupt handler from acting on information that is only partially updated.

The **terminate** routine turns off the vertical interrupt, and restores the vertical interrupt vector to its original value and the control bit for IRQ2 in the interrupt control register at port 21H to its original state. As discussed above, the restoration of IRQ2 to its original state is particularly important in the AT, because failure to restore it could result in a nonfunctional hard disk.

The initialization, update, and termination routines are merely support for the heart of the sprite engine—the vertical interrupt handling routine **put_objects_on_screen**, which is executed whenever an IRQ2 interrupt occurs. The first action of this interrupt handler is to determine whether the IRQ2 interrupt was in fact generated by the EGA, by checking bit 7 of the status register at port 3C2H. If this bit is 0, the EGA did *not* generate the interrupt and control must be passed to the original interrupt 0AH vector, as stored by the initialization routine. Again, this action is essential in the AT.

If the interrupt was generated by the EGA, the interrupt handler increments **vert_int_modulo_count**, a public variable that may be accessed by the application program. A change in this variable notifies the application that a vertical interrupt has occurred, allowing implementation of a constant time base synchronized to the frame rate of the EGA. The interrupt handler then checks to see if any of the objects have changed, as would be indicated by **need_to_draw_something_flag**. Animation activity is required only if at least one object has changed. If nothing has changed, the interrupt handler issues a general end-of-interrupt (EOI) command to reenable IRQ2, clears and reenables the EGA bits that control the vertical interrupt, and restores the registers that it modified. Interrupt handlers must preserve all registers, because they may be invoked if interrupts are enabled. This can happen during the execution of any code. The interrupt handler then terminates immediately; as a result, very little overhead is incurred when none of the objects has changed since the last vertical interrupt.

If one or more objects indeed has changed, the interrupt handler must erase and then redraw all of the objects in the queue. To erase all objects at the location at which they were last drawn, the interrupt handler executes a loop that calls a precalculated entry point in in-line code for each object; the in-line code transfers the rectangular area of the background corresponding to the location of the object from the background buffer to display memory. This erases the object, by restoring the area it occupies to the background state.

One special feature of this queue is that it does not erase the object under certain circumstances. If, for example, the object has not moved since the last time it was drawn, and if the object's image is the same as when it was last drawn, then erasing the object is not necessary, because it will be drawn over exactly the same pixels that would be erased. This provides one way to support an extra object or two during the limited vertical retrace period: if the application can guarantee that only half of the objects being animated change during any given vertical frame, then the erase time for half the objects is always saved, freeing up drawing time for additional objects. Also, if the queue contains the special 0FFFFH entry for the offset at which the object was last drawn, the object is not erased. This can save time when an object is made to vanish, because an invisible object does not need to be drawn or erased.

After all objects have been erased at their old locations, they are redrawn at their new locations. This is accomplished in a second loop using the AND/OR driver. Like the erase driver, the AND/OR driver in-line code is called directly via a vector precalculated for each object in the update routine.

The implementation of the AND/OR driver shown in listing 2 uses word operands for maximum speed. For an image that is an odd number of bytes wide, an extra mask byte of 0FFH and image byte of 00H can be added to the end of each line of the image table, with no adverse effects because this pair of bytes causes the AND/OR driver to leave display memory unchanged. The driver also uses string instructions as much as possible, but because the AND and OR logical functions have no string equivalents, the driver must rely largely on slower 8088 instructions. This driver is not nearly as fast as the byte-move driver in listing 3, but it is remarkably speedy considering its sprite-like characteristics.

Like erase, the redraw loop does not draw objects with a screen offset of 0FFFFH. This provides a handy way to disable an object. The method works as follows: if the application calls the up-

XENIX RISING

"It was back in 1986 that we first grasped the full significance of **SCO XENIX**®.

"We had been thinking of it as just an 'operating system'—as we used to call them—when it was really much more. It was the foundation of a whole new approach to shared information and resource computing for PCs: networked DOS *and* XENIX workstations.

"The '**SCO XENIX** solution' integrated XENIX and DOS, multiuser and LAN, and PCs and mainframes, into a unified environment unprecedented in its power, productivity, and price performance per user for personal computers.

"SCO's solution included not only 16-user licensed **SCO XENIX** itself, but also **SCO XENIX-NET** XENIX/DOS networking; **SCO Professional**™ and **SCO FoxBASE**™, SCO's XENIX-based workalikes of 1-2-3® and dBASE II™; **SCO Lyrix**™ Word Processing System, and other productivity tools; and **SCO uniPATH**™ SNA-3270 Mainframe Communications, plus languages and graphics packages.

"Together with unparalleled SCO hot-line support, documentation, and training, plus the explosive growth of the XENIX applications base, it changed the way we would look at personal computers forever.

"In short, it was the '**SCO XENIX** solution' that turned personal computers into *real* computers."

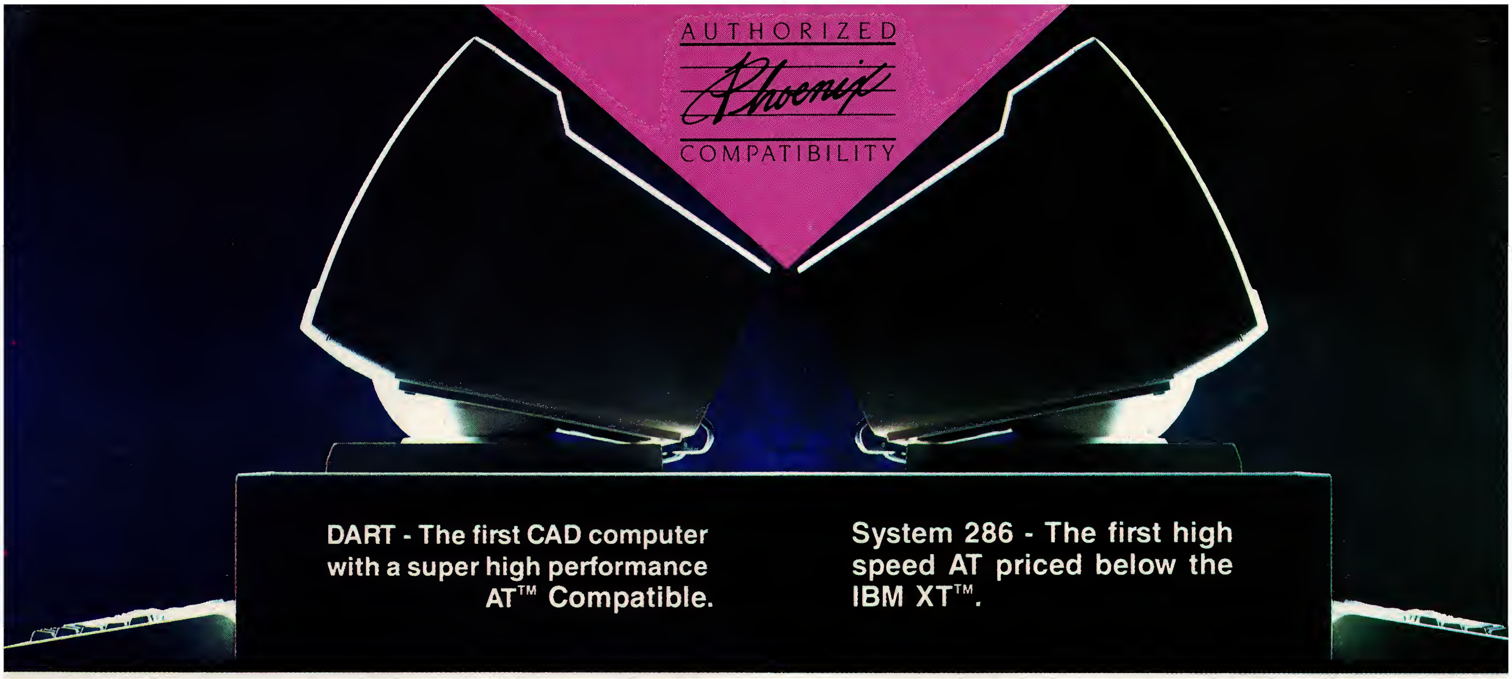
from *"XENIX RISING: The Ascent of Shared Information Computing from 1979 to 1999"*

SCO SALES AND INFORMATION
(800) 626-UNIX
(800) 626-8649
(408) 425-7222
TWX: 910-598-4510 SCO SACZ
uucp: ...decvax!microsoft!sco!info

SCO
THE SANTA CRUZ OPERATION

SCO XENIX System V is available for IBM® PC AT®, PC XT®, ITT XTRA® XL and other compatibles, and AT&T PC 6300 Plus–now!

CIRCLE NO. 121 ON READER SERVICE CARD



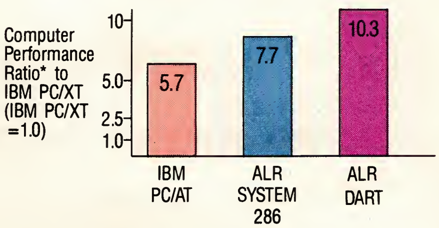
The ADVANCED LOGIC RESEARCH computer has been tested for IBM compatibility with PHOENIX'S compatibility test suite using the PHOENIX AT-COMPATIBLE ROM BIOS, and the PC-COMPATIBLE ROM BIOS.

WE DON'T THINK YOUR PERFORMANCE SHOULD BE LIMITED BY THE 'STANDARD'

At ALR we don't think your performance should be limited by the 'Standard'. So preoccupied are today's computer makers with meeting the 'Standard', they seem to have overlooked the opportunity to do something much more worthwhile.

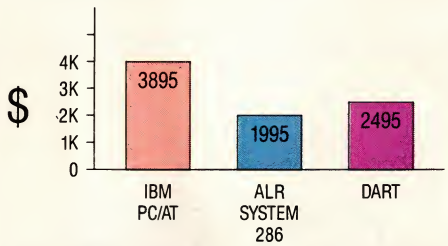
Namely, the concept of improvement beyond that 'Standard' acceptable level.

Examine the DART. The first CAD computer with a super high performance AT™ Compatible. The DART system is powered by a 10 MHz 80286 cpu with support for an 8 MHz 80287 math co-processor, when compared to the IBM® AT's 6 MHz cpu and 4 MHz co-processor, the DART system will process your next CAD design in almost half the time, with DIRECT ACCESS RESPONSE TIME (DART).



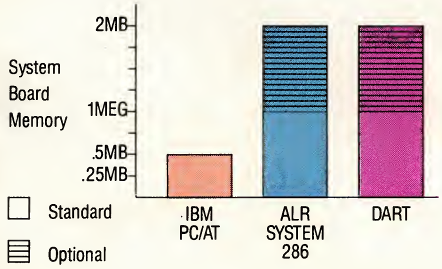
*Based on the Norton Utility SI-System Information Command.

Explore the opportunity of the System 286 - The first high speed AT below the price of an IBM XT™. The low cost of this system did not reduce the performance, with the cpu speed of 8 MHz and co-processor speed of 5 MHz. This System 286 is ready to calculate spread sheets and sort data base in a flash!

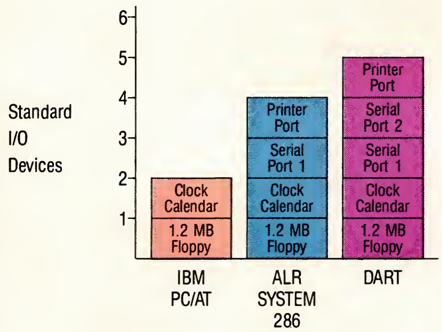


LIST PRICES AS OF APRIL 1986

The concept of having enough memory is misleading. It seems yesterday's 64K of memory was enough. Today, computers require at least 512K in order to run popular programs such as Symphony™ or Framework™. The result? ALR System 286 and DART system both have four times the memory capability built into its 'motherboard' so you don't have to pay for it tomorrow.



Option I/O device? That usually implies you need it and it costs more. ALR believes a computer needs to interface with at least a printer and in most cases modems, plotters and other computers, so we design our systems to include more standard I/O.



High performance and more features usually results in a more expensive system. However, the ALR System 286 and DART system does not match the 'Standard' price and this is one category we're glad to be considered 'sub-standard'.

Let one of our sales engineers or dealers show you the ALR System 286, DART system, Turbo XT and AT Hirise IBM compatible systems that are designed to limit your cost not your standard of performance.

CIRCLE NO. 199 ON READER SERVICE CARD

ADVANCED LOGIC RESEARCH, INC.
10 CHRYSLER
IRVINE, CA 92714
(714) 581-6770
ALR IN CANADA
416-229-6477

ALR

ENGINEERING ADVANCED COMPUTERS

date routine with SI equal to 0FFFFH, the routine sets the screen offset for that object to 0FFFFH. Upon encountering this object, the erase loop erases it; the new offset of 0FFFFH is guaranteed to be different from the offset of the old location. The redraw loop does not redraw the object because it ignores objects with an offset of 0FFFFH. Thus, the object vanishes.

The erase loop also ignores objects with screen offsets of 0FFFFH, so the object not only vanishes, but it is never erased again and requires little overhead. This makes it possible to change the relative drawing priorities of objects; a single object can be assigned several priorities (queue numbers), all but one of which is set to 0FFFFH at any one time. To make the object appear far forward, a high-priority (high-numbered) queue entry would be activated; to make the object appear far to the rear, a low-priority (low-numbered) queue entry would be turned on. In combination with the use of larger images for the more forward versions of the object to simulate perspective, priority-switching can produce very convincing animation effects.

Once the objects have been erased and redrawn, the vertical interrupt handler is done. The interrupt handler issues a general EOI command to reenable IRQ2, clears and reenables the vertical interrupt, restores all 8088 registers, and terminates.

An important note about the vertical interrupt handler: interrupts are enabled soon after the handler is entered and are kept enabled throughout. If another, higher-priority interrupt, such as the timer interrupt, occurs while the vertical interrupt handler is active, the vertical interrupt handler will be halted for an indeterminate period. Because the visual effect of the sprite engine is dependent upon its finishing during the vertical retrace period, this would seem to pose a significant problem, but actually it does not. If an interrupt does cause the sprite engine to run over into display time, the worst that will happen is a single flicker; if the interfering interrupt is handled quickly, no visible effect may be detected at all, although a frequent and long interrupt could cause problems. Also, IBM apparently has never specified a maximum allowable period for which interrupts can be disabled by an interrupt handler; in the absence of such a specification, interrupt-off periods always should be kept as brief as possible.

Note that only the graphics driver in-line code portion of the sprite engine has been optimized. This code absorbs most of the time used by the engine during vertical retrace, and it is only during this period that time is critical. Besides, at levels above the driver level, clarity and flexibility are as important as sheer speed, except when the application dictates otherwise.

APPLYING THE SPRITE

The application program sets up and drives the sprite engine to animate the helicopters and the balloon. This program comprises three parts: initialization, animation, and termination.

The initialization section selects medium-resolution graphics mode, places the stars and the horizon in the background buffer, and uses the byte-move driver shown in listing 3 to put the moon in the background buffer. (To make the byte-move driver draw into the background buffer instead of display memory, ES is set to the background buffer segment.) The completed background buffer is then copied into display memory to establish the background. The initialization code concludes by calling the sprite engine initialization routine, which sets up the queue and the vertical interrupt.

The animation section performs only three actions: animation control, check for exit, and synchronization. The program first loops through each object, checking whether the delay for each object has counted down. If it has, it moves the object based on its current motion increment, then cycles to the next form table on the list of form tables for that object, in order to produce internal animation. (The helicopter's blades and tail rotors move through a process of internal animation.) If the repeat factor for the current motion has counted down, then the next set of control variables is obtained from the control string for that object.

The control strings, which are **copter0_control**, **copter1_control**, and **balloon_control**, fully describe the motion and appearance of the objects throughout program execution, so changing the animation of an object is a simple matter of altering the appropriate string. For example, it may be instructive to set the delays in the control strings to higher values to slow down the program in order to more closely observe the effects when objects cross.

When an object is moved, a call is made to **object_services**, the sprite engine's update routine, to register the change. This done, the application has provided the sprite engine with all the information it needs to do its work and

is free to perform other tasks. The application then checks for a keystroke. If it finds one, the program is terminated immediately; if a keystroke is not found, the program synchronizes to the EGA's frame rate by waiting for the variable **vert_int_modulo_count** to change. Recall that the sprite engine increments this variable each time a vertical interrupt occurs, so waiting for it to change forces the execution speed of the applications program to exactly 60 iterations per second, matching the 60 frames per second rate of the EGA. This provides the program with a constant time base; at all times and under all circumstances, this program will animate at the same speed, with a control resolution of one-sixtieth of a second. This is true even on an AT, regardless of clock speed.

When a keystroke is detected, which indicates program termination, the sprite engine's termination routine is called to restore the vertical interrupt to its original state, text mode is restored, and the program is ended.

The key aspect of the sprite engine application is the degree to which its use reduces the code and complexity required to implement effective animation. The actual animation code is only a couple of dozen lines long. Yet those lines support a powerful and extensible table-driven animation interpreter. Amazingly, this program is not very far from being a full-fledged game.

CGA IMPLEMENTATION

The sprite engine can be implemented for the CGA as well, although less elegantly. When running on the CGA, the application must emulate the vertical interrupt by waiting for the vertical synchronization pulse and then calling the sprite engine. The problems associated with this approach have been discussed.

To assemble the sample program for the CGA, change the equates for the **ega** variable at the start of listings 1 and 2 to 0, rather than 1, thus enabling the conditional assembly blocks that start with **if cga** and disabling those that start with **if ega**. The CGA version will run equally well on an EGA because the EGA without the vertical interrupt enabled is the same as the CGA (for the purposes of this program). A review of the two versions in quick succession, however, reveals the significant difference that exists between them.

The balloon often flickers when it reaches the top of the display in the CGA version. (This effect occurs only on 4.77-MHz PCs; faster machines should produce no flicker.) This effect is almost expected because insufficient time is available to draw all the objects during the vertical retrace period, the result of invoking the sprite engine at the beginning of the vertical synchronization pulse rather than the beginning of the vertical retrace period. However, some far more profound lessons can be learned from this event: one pertains to the nature of realtime bit-mapped animation, the other to a method of maximizing the capabilities of the sprite engine through program structure rather than code optimization.

(As an aside, note that the balloon flickers only when the green helicopter is visible. Remember that when an object is invisible, it is neither erased nor redrawn. This is a visual reinforcement of the concept that the balloon flickers because too little time is available to erase and redraw three objects during the period that is used.)

ANIMATION DYNAMICS

It is natural to think of the time during which the sprite engine can safely modify display memory as being static. Given that the last 60 scan lines of each

frame are in the vertical retrace period, it seems obvious that the time during which those lines occur is the time available for modifying display memory without the results being displayed until the operation is complete. This is the truth, but not the whole truth.

Actually, a given area of display memory can be modified at any time during which the display controller is not fetching that particular area of memory to convert it to video data. Any operation that is begun after the controller has read the area of display memory to be modified, and finished before the controller completes a frame and returns to read the affected memory again, will not cause flicker. Completing all operations during the vertical retrace period is merely a special case of modifying memory between successive passes of the display controller.

The exceedingly dynamic nature of realtime bit-mapped animation is evident in the manner in which the balloon vanishes. The expectation might be that during those frames when the balloon flickers (the frames during which the balloon is still erased when the display controller encounters it), the balloon would vanish smoothly, two scan lines (the increment of its vertical motion) at a time, as if passing behind a dark wall, as it rises to the point at which it encounters the display controller scan (and by extension the scanning electron beam on the display). To the contrary, the balloon vanishes from the top in two large chunks and is gone. (To see this effect, force the sprite engine to redraw all forms on every vertical interrupt by making two changes to **put_objects_on_screen**: replace the line **je process_queue** with **jmp process_queue**, and comment out the six lines that start with **je skip_this_object** and end with the next instance of **je skip_this_object**.)

Moreover, it takes some reflection to understand why it is the top of the balloon that vanishes first. After all, the balloon is drawn from the top down; it is reasonable, therefore, to assume that the portion of the balloon that is drawn first should be visible after the rest of the balloon has vanished.

The explanation lies in the interaction of the display controller and the graphics driver. Both are scanning display memory in much the same fashion and at much the same speed. The controller is scanning one line from left to right every 62.5 microseconds, progressing downward. At the same time, the graphics driver is drawing the balloon from left to right, progressing downward. While the exact time per scan line of the balloon is difficult to figure out, an approximation arrived at by adding instruction cycle times is 73 cycles per word, or somewhere around 200 cycles for each of the two-word-wide balloon scan lines, including overhead, wait states, and prefetch queue effects. This works out to 42 microseconds, or two-thirds of the time the display controller takes per scan line.

The balloon vanishes the way it does because the graphics driver actually starts drawing the balloon in memory that the display controller has already scanned past during the current frame. However, the driver is only slightly behind the controller, and, because it is faster, the driver overtakes the controller. Up to this point, nothing drawn by the graphics driver is visible, because that memory was erased when the display controller read it. Consequently, the top of the balloon vanishes.

Soon the graphics driver passes the display controller and starts to modify memory just before that memory is read for video data. At this point, the

bytes drawn by the driver become visible, because the data are there in time to be read by the display controller. As a result, the lower part of the balloon is visible on the screen.

But why does the balloon disappear in two large chunks, rather than two scan lines at a time? The answer is that the graphics driver takes about three scan lines to catch up to one scan line on the display controller; therefore, when the balloon moves up two scan lines, a half dozen or so additional scan lines vanish as the graphics driver takes longer to catch up. Note, however, that it is mere coincidence that the balloon is just the right size and the driver the right speed to produce this effect. When a helicopter is substituted for the balloon, for example, it vanishes from the bottom up, rather than the top down. Because each helicopter scan line is three words long, not two, it takes slightly longer to draw a helicopter scan line than the display controller takes to scan a line, thus, the controller catches up with the driver rather than the other way around.

The variables in animation, then, cannot be considered in isolation from each other, for they are synergistic and the effects of their interaction are not always obvious or predictable.

ENHANCED ANIMATION

The following are two examples of superior animation. One draws on an understanding of the dynamic nature of animation, the other highlights the illusory nature of animation in general.

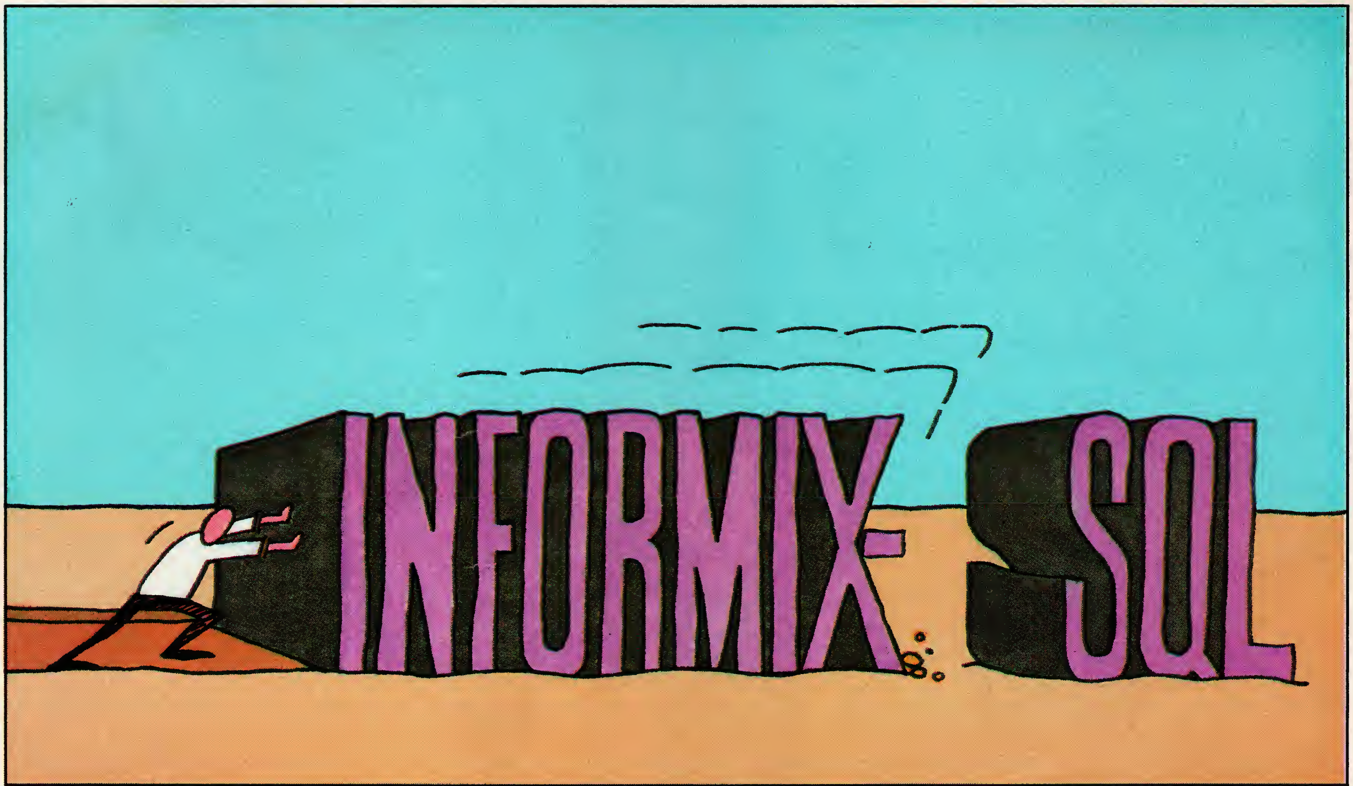
First, suppose a user wanted to add one or two more objects to the animation program. The CGA version certainly will not support any more objects; the EGA version does not have much surplus animation time, either. Yet, with a few restrictions, another object or two can be added and the CGA's balloon problems cured, all at once.

The balloon's tendency to vanish can be solved by simply not letting the balloon rise as high as the point at which it starts to disappear. Cheating? Not really. The objective in animation is to produce the desired effects within the limitations of the host hardware, and unless some compelling reason exists as to why the balloon must rise all the way to the top of the screen, the game designers could just make the very top of the screen the scoreboard, or the boundary line, or just blank space. (In many games for the Apple II, for example, the far right of the screen was not part of the playing field—often occupied instead by cheering creatures or the scores—because that area of the screen could not be addressed with a single-byte column coordinate.)

Recall also that the red helicopter is actually drawn after the balloon; it appears to be the most forward of the objects. In the CGA version, then, the red helicopter is drawn entirely during vertical display time. Yet this helicopter does not flicker or vanish at all, an effect that would be understandable given the examination of realtime animation dynamics above.

The effect achieved is the trick needed to add a few extra objects to the animation: objects are added so the forward-most objects are restricted to the lower portion of the screen. This ensures that the objects drawn last—during display time—are drawn ahead of the display controller's scanning of memory. Display time takes up about three-quarters of the total time, providing considerable latitude in adding objects, as long as the program is structured to keep the drawing of all objects ahead of the display controller. Remember that all objects are erased before any redrawing takes place, so the addition of an object inevitably forces all objects to be drawn a little later relative to the start of vertical retrace.

An alternative means of supporting additional objects would be to reduce



How we improved Structured Query Language.

Actually, we didn't change a thing.

We just combined it with the best relational database management system.

Introducing INFORMIX®-SQL.

It runs on either MS™-DOS or UNIX™ operating systems. And now with IBM's SQL as part of the program, you can ask more of your database. Using the emerging industry-standard query language.

To make your job easier, INFORMIX-SQL comes with the most complete set of application building tools. Including a full report writer and screen generator. Plus a family of companion products that all work together.

Like our embedded SQLs for C and COBOL. So you can easily link your programs with ours. File-it!,™ our easy-to-use file manager. And C-ISAM,™ the de facto standard ISAM for the UNIX operating system. It's built into all our products, but you can buy it separately.

And when you choose RDS, you'll be in the company of some other good companies. Computer manufacturers including AT&T, Northern Telecom, Altos and over 60 others. And major corporations like Anheuser Busch and The First National Bank of Chicago.

Which makes sense. After all, only RDS offers a family of products that work so well together. As well as with so many industry standards.

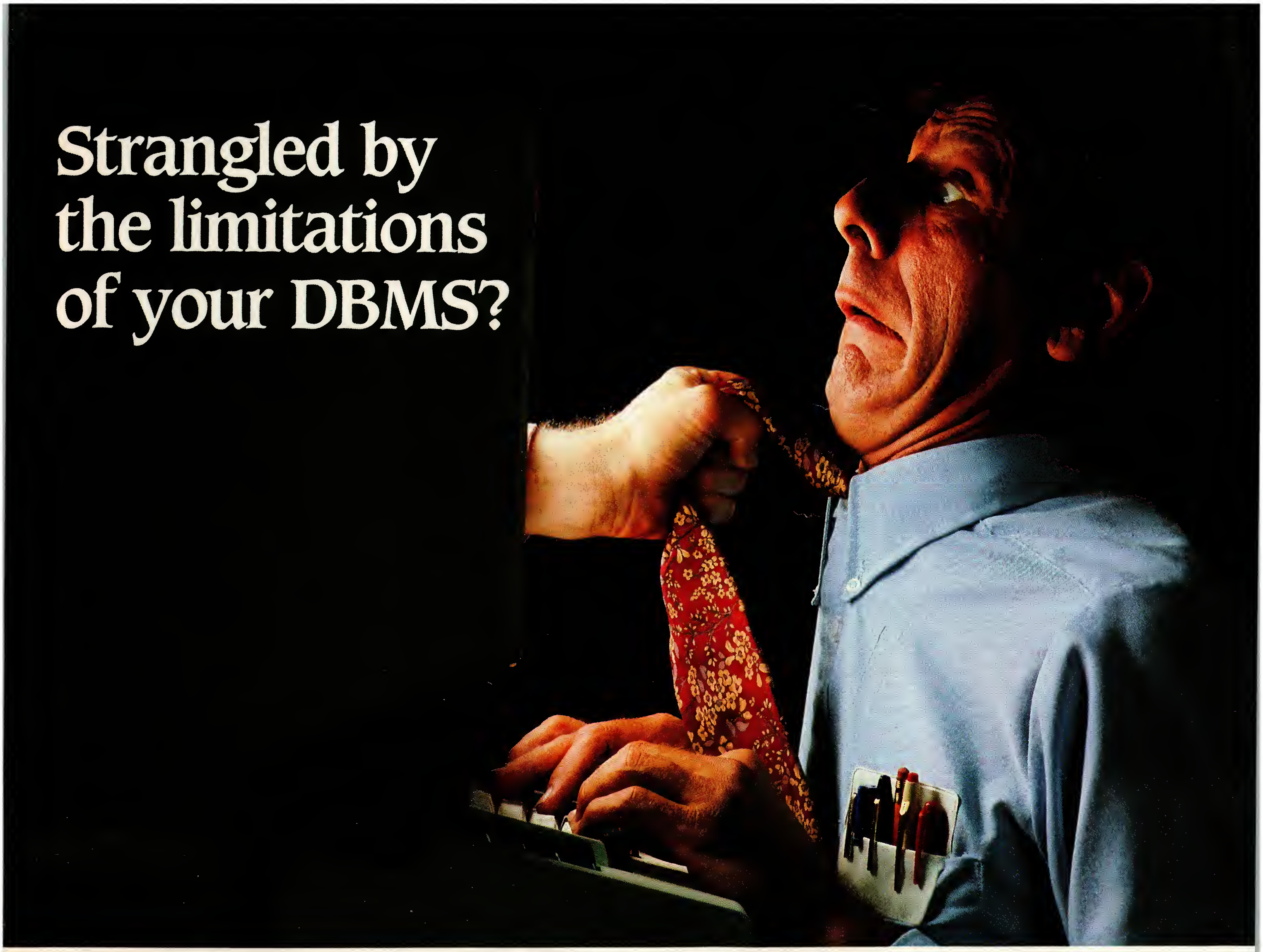
So call us for a demo, a manual and a copy of our Independent Software Vendor Catalog. Software vendors be sure to ask about our new "Hooks" software integration program. Our number: 415/322-4100.

Or write RDS, 4100 Bohannon Drive, Menlo Park, CA 94025.

And we'll show you how we took a good idea and made it better.

RDS™

RELATIONAL DATABASE SYSTEMS, INC.



ZIM. The system for serious developers

Your data base system was easy to learn. You were building your first simple applications within hours. You have a menu driven interface and pop up menus with online context sensitive help. Now you are ready to build your first real application. Instant Strangulation!

The Symptoms

- There is no real 4th Generation Language – what you see at the menu driven level is what you get.
- There are all sorts of built-in limitations which no real problem ever obeyed
- Your database which runs like a superstar on a few files and 500 records, chokes when faced with several files of 50,000 records
- You cannot efficiently run your applications in UNIX, VAX VMS and MS-DOS without major surgery, or finding a new DBMS
- Your DBMS supplier asks, "what is networked multiuser anyway?"
- Your database is too big to live on one disk but your DBMS doesn't support multiple disks
- Your database can't answer simple questions like "tell me which inventory items have **NOT** sold in the past 3 months?"
- You built your application using the nice easy interface, but now it won't get out of the way to allow you to modify the application
- You can't compile your applications and there is no runtime system

Diagnosis

Extreme stress and frustration in your development staff and applications crippled at birth.

THE CURE

ZIM is the 4th Generation Language/ DBMS for serious application developers. ZIM is a top performer in all of its environments (UNIX, XENIX, MS-DOS, Novell Networks, VAX VMS, QNX), for all types of information application. ZIM's easy menu driven interface with context sensitive help is not meant just for novice users, and it stays discreetly out of the way when it isn't needed.

You need never suffer the symptoms of Database strangulation again.

ZANTHE INFORMATION INC.
1200-38 Antares Drive
Nepean, Ontario, Canada K2E 7V2
Sales (US) 800-267-9972 (CDN) 613-727-1397

ZANTHE INFORMATION INC

ZIM™

CIRCLE 219 ON READER SERVICE CARD

the average size of the objects. Again, the proper balance between the hardware's capabilities and the needs of the application is the key.

A second example of enhanced animation is a program structure that supports an arbitrarily large number of objects. Observe that no matter what tricks are played with the sprite engine discussed above, the limitation on the total number of objects that can be animated is the number that can be erased and redrawn during a single frame. However, this is true only as long as all objects can overlap. If the motion of the objects is restricted such that any object can overlap only with objects within a certain group, and the periodicity with which the groups of potentially overlapping objects need to be redrawn is arranged so that only one group ever needs to be redrawn during any given frame, then dozens of objects can be animated. Only as many objects as can be redrawn within a single frame can be in any group (as discussed above), but the interleaving on a frame-by-frame basis of the redrawing of the groups makes the illusion of simultaneous animation easy to create.

An obvious candidate for such animation is a shooting gallery game, where the many rows of targets do not overlap. No one would ever notice in a frenzied shoot-'em-up game that only certain subsets among dozens of moving objects ever actually cross. It is all illusion, and it is the organization of the animation that makes the illusion work.

FUTURE FLEXIBILITY

The programs presented here provide the user with groundwork that can expand in many directions. The sprite engine could be designed with expanded functionality to relieve the application of additional work. For example, the sprite engine could be provided with the address of the movement control vector for each sprite, then it could move the sprites independently of the application. The sprite engine also could assume responsibility for collision detection, because it is already scanning the objects frequently anyway.

The code used to control the movements of the sprites in listing 1 is a very simple interpreter. Any degree of sophistication can be introduced simply by adding (for each sprite) a queue entry that is a vector to a handling routine for that sprite. The sprite-handling routines could include movement, collision detection, event generation, and other functions. Many methods of implementing a powerful animation interpreter are available. The important point is that with the proper design, an animation interpreter can be flexible and extensible, so that adding features requires only the modification of the control tables and perhaps the addition of a couple of routines.

The sprite engine can be used in conjunction with other animation techniques. In a shooting gallery game, for example, the primary object (perhaps a spaceship) might never intersect other objects without wiping them off the screen (or vice versa). In this case, the primary object need not be drawn by the sprite driver, because no possibility of overlap exists. It also may be desirable to move the primary object faster and with more flexibility than the other objects; this may be a good application for a faster byte-move driver operating in the foreground, with the sprite engine running independently in the background. No driver is best for all applications. Look to innovative program structure and the proper mix of animation techniques for the best results.

When considering potential animation applications, remember that results can be much different with other 8086-family processors. The 80286 processor, in particular, has great potential for animation applications. The sprite engine described here was designed for the slowest CPU, the 8088, because it is also the most common. Drivers reoptimized for the 80286 could support two to three times as many objects.

Finally, be aware that objects have been horizontally byte-aligned rather than pixel-aligned, for the sake of simplicity and reduced program length. Pixel alignment, along with smaller movement increments for the objects, would provide smoother motion and allow more finely detailed internal animation; it is strongly recommended for serious animation applications.

This sprite engine is a standardized, fully functional solution to the animation problems that were created by IBM's decision not to include animation hardware in the PC. This engine provides visually superb animation while shielding programmers from its details, thereby expanding their perspective in crafting application code, drivers, and hardware into a smoothly working whole. This surely can result only in better animation software.

Michael Abrash is a senior software engineer for Tseng Laboratories in Pennsylvania. Dan Illowsky is president of Funtastic. Together and separately they have written video games for the IBM PC and Apple II.

LISTING 1: SPRITES1.ASM

```
;
;This program demonstrates the use of a set of routines which
; allow smooth, non-flickering, and non-destructive animation
; of objects on an IBM CGA or EGA graphics adapter. The program
; moves two helicopters and a balloon along programmed paths while
; internally animating the helicopter's roters, and orientation. You
; will see no undesirable fringe or overlap effects. Objects look
; as if they are in separate planes, and blank areas within an object
; such as the windows of the helicopters appear to be transparent.
;
; Link with Listings 2 and 3 with the command:
;       link sprites1+sprites2+sprites3;
; to generate the executable sprite application file sprites1.exe.
;
stack   segment para stack 'STACK'
        db      100 dup('Stack')
stack   ends
;
;Segment used to hold memory image of the background. It is used to
; restore the background before an object is moved.
;
background_segment  segment para 'DATA'
        db      4000h-((4000h-16000)/2) dup(0) ;last 192 bytes of the
background_segment  ends                        ; 16K video memory
;                                               ; buffer are never used
cseg    segment para public 'cseg'
        assume  cs:cseg,ds:cseg,es:nothing
        extrn   byte_move_form_driver:near
        extrn   initialize:near,terminate:near,object_services:near
        extrn   vert_int_modulo_count:word
        extrn   put_objects_on_screen:far
;
;The flag below must be set properly before assembling this program
;
ega     equ  0 ;1 to assemble for use on an Enhanced Graphics Adapter
               ;0 to assemble for use on a Color Graphics Adapter
cga     equ  (ega xor 1) ;the opposite status of ega
;
if ega
old_vert_count  dw      0       ;holds number used to synchronize this
                                ; program to vertical interrupt
endif
;
screen_buffer_adr       equ     0b800h      ;video mode 4's buffer
                                            ; paragraph address
;
num_objects     equ     3       ;this program will move three objects
;
;Below is a table which holds all the parameters which govern the
; movement of the three individual objects
x           dw              15,             38,             64
y           dw              90,            180,            164
xinc        dw               1,              0,             -1
yinc        dw               0,             -2,              0
form        dw     bcptr_rt_tbl,    balloon_tbl,    rcptr_lft_tbl
formbase    dw     bcptr_rt_tbl,    balloon_tbl,    rcptr_lft_tbl
formmax     dw bcptr_rt_tbl_max, balloon_tbl_max, rcptr_lft_tbl_max
control     dw    copter1_cinit,   balloon_cinit,   copter0_cinit
controlbase dw  copter1_control, balloon_control,  copter0_control
controlmax  dw copter1_cntl_max, balloon_cntl_max, copter0_cntl_max
delay       dw              01,             01,             01
delaybase   dw              04,             12,             06
repeat      dw              47,             91,             55
;
;Below are lists of parameters which control the sequence of
; parameters used to define the movement patterns, and shapes.
; The parameters are listed in the following order:
;       xinc, yinc, form, formmax, delay, repeat
;

copter0_control dw -1,0
                dw rcptr_lft_tbl,rcptr_lft_tbl_max,06,54
copter0_cinit   dw 0,-2
                dw rcptr_lft_tbl,rcptr_lft_tbl_max,06,35
                dw 1,0
                dw rcptr_rt_tbl,rcptr_rt_tbl_max,06,4
                dw 1,2
                dw rcptr_rt_tbl,rcptr_rt_tbl_max,08,10
                dw 1,-2
                dw rcptr_rt_tbl,rcptr_rt_tbl_max,08,8
                dw 1,-2
                dw rcptr_rt_tbl,rcptr_rt_tbl_max,08,4
                dw 0,0
                dw rcptr_rt_tbl,rcptr_rt_tbl_max,10,10
                dw 1,-2
                dw rcptr_rt_tbl,rcptr_rt_tbl_max,08,4
                dw 1,-2
                dw rcptr_rt_tbl,rcptr_rt_tbl_max,08,9
                dw 1,2
                dw rcptr_rt_tbl,rcptr_rt_tbl_max,08,15
                dw 0,2
                dw rcptr_rt_tbl,rcptr_rt_tbl_max,10,35
copter0_cntl_max label   byte
;
copter1_control dw 1,0
                dw bcptr_rt_tbl,bcptr_rt_tbl_max,04,46
copter1_cinit   dw -1,0
                dw bcptr_lft_tbl,bcptr_lft_tbl_max,04,46
                dw 1,0
                dw bcptr_rt_tbl,bcptr_rt_tbl_max,04,46
                dw -1,0
                dw bcptr_lft_tbl,bcptr_lft_tbl_max,04,46
                dw 0,0
                dw disappear_table,disappear_table_max,300,1
copter1_cntl_max label   byte
;
balloon_control dw      0,-2
                dw      balloon_tbl,balloon_tbl_max,12,70
balloon_cinit   dw      0,2
                dw      balloon_tbl,balloon_tbl_max,12,70
balloon_cntl_max     label   byte
;
;Below are tables used to sequence the internal animation
;
rcptr_lft_tbl       label   word
        dw      rcopter_left0
        dw      rcopter_left1
rcptr_lft_tbl_max  label    word
;
rcptr_rt_tbl      label   word
        dw      rcopter_right0
        dw      rcopter_right1
rcptr_rt_tbl_max label   word
;
bcptr_lft_tbl       label   word
        dw      bcopter_left0
        dw      bcopter_left1
bcptr_lft_tbl_max  label   word
;
bcptr_rt_tbl      label   word
        dw      bcopter_right0
        dw      bcopter_right1
bcptr_rt_tbl_max label   word
;
balloon_tbl             label   word
        dw      balloon
balloon_tbl_max         label   word
;
disappear_table             label   word
        dw      0ffffh
disappear_table_max         label   word
;
;Below are the forms used to define the various images of the objects
; in a form suitable for an AND-OR type form driver. The format is:
;       byte 0 - height in lines (h)
;       byte 1 - width in bytes  (w)
;       followed by w X h (mask word, image word) pairs.
;
rcopter_left0   label   byte
        DB  00Eh, 006h, 000h, 000h, 0FFh, 0FFh, 000h, 00Fh, 0FFh, 0F0h
        DB  0FFh, 0FFh, 000h, 000h, 0FFh, 0F0h, 000h, 00Fh, 000h, 00Fh
        DB  0FFh, 0F0h, 0FFh, 0FFh, 000h, 000h, 0FFh, 0F0h, 000h, 00Fh
        DB  0FFh, 0FFh, 000h, 000h, 0FFh, 0FFh, 000h, 000h, 0FCh, 000h
        DB  002h, 0AAh, 003h, 0FFh, 0A8h, 000h, 0FFh, 0FFh, 000h, 000h
        DB  0F3h, 0FFh, 008h, 000h, 0C0h, 0FFh, 02Ah, 000h, 0FFh, 03Fh
        DB  000h, 0C0h, 0CFh, 0FFh, 020h, 000h, 0F0h, 03Fh, 00Ah, 080h
        DB  0FFh, 03Fh, 000h, 0C0h, 03Fh, 0FFh, 080h, 000h, 0F0h, 00Ch
        DB  00Ah, 0A2h, 03Fh, 03Fh, 080h, 0C0h, 03Fh, 0FFh, 080h, 000h
```

MDBS III®

ATTN: APPLICATION DEVELOPERS

. . . DELIVERS THESE ESSENTIAL FEATURES. DOES YOUR DBMS?

MDBS III is more powerful than most mainframe data base management systems. . . and less expensive. MDBS III was designed for serious application developers like you. Like the developers of Solomon III, the "Number One" accounting system. And all the others who demand these essential features MDBS III provides:

DATA STRUCTURING—So flexible it captures any data relationship you can imagine. So comprehensive you'll design complex data bases faster than ever.

TRUE MULTI-USER—Few DBMSs give you as many facilities to guard against haphazard concurrent data modification as MDBS III does, down to the locking of individual data records.

PERFORMANCE—MDBS III gives you fast data modification and retrieval plus extensive performance tuning facilities.

DATA INTEGRITY—MDBS III provides airtight integrity assurances. . . from range checking to transaction-logging to enforcement of data relationships. . . *all automatically.*

PHYSICAL DATA PROTECTION—You get automatic recovery from media as well as from physical data destruction.

DATA SECURITY—Protect your data using passwords, encryption, and read/write access down to the field level.

PORTABILITY—MDBS III runs on a range of mini and micro computers, including LANs, and supports a variety of host language interfaces.

SUPPORT—**mdbs** is there when you need us, with in-depth seminars, telephone support, individual consulting and contract programming to help you develop and install your applications.

Call us today at 800-323-3629 for more information; in Canada or Illinois, dial 312-303-6300. Or write **mdbs**, P.O. Box 248, Lafayette, IN 47902, TELEX 209147 ISE UR.

MDBS III®
ABSOLUTE POWER

CIRCLE NO. 211 ON READER SERVICE CARD

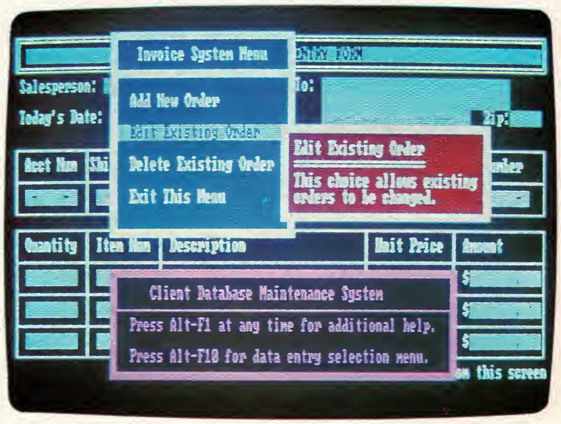
mdbs®

```
        DB  0F0h, 003h, 00Ah, 0A8h, 0C0h, 000h, 03Fh, 0FFh, 03Fh, 0FFh
        DB  080h, 000h, 0C0h, 00Ch, 02Ah, 0A2h, 03Fh, 03Fh, 080h, 0C0h
        DB  000h, 000h, 0AAh, 0AAh, 000h, 03Fh, 0AAh, 080h, 0FFh, 03Fh
        DB  000h, 0C0h, 0C0h, 000h, 02Ah, 0AAh, 000h, 0FFh, 0AAh, 000h
        DB  0FFh, 03Fh, 000h, 0C0h, 0F0h, 000h, 00Ah, 0AAh, 003h, 0FFh
        DB  0A8h, 000h, 0FFh, 0FFh, 000h, 000h, 03Ch, 0FFh, 0C3h, 000h
        DB  0CFh, 03Fh, 030h, 0C0h, 0FFh, 0FFh, 000h, 000h, 0C0h, 000h
        DB  03Fh, 0FFh, 000h, 0FFh, 0FFh, 000h, 0FFh, 0FFh, 000h, 000h
rcopter_left1   label   byte
        DB  00Eh, 006h, 000h, 000h, 0FFh, 0FFh, 000h, 00Fh, 0FFh, 0F0h
        DB  0FFh, 0FFh, 000h, 000h, 000h, 000h, 0FFh, 0FFh, 0FFh, 0FFh
        DB  000h, 000h, 0FFh, 0FFh, 000h, 000h, 0FFh, 0F0h, 000h, 00Fh
        DB  0FFh, 0FFh, 000h, 000h, 0FFh, 0FFh, 000h, 000h, 0FCh, 000h
        DB  002h, 0AAh, 003h, 0FFh, 0A8h, 000h, 0FFh, 0FFh, 000h, 000h
        DB  0F3h, 0FFh, 008h, 000h, 0C0h, 0FFh, 02Ah, 000h, 0FFh, 0FFh
        DB  000h, 000h, 0CFh, 0FFh, 020h, 000h, 0F0h, 03Fh, 00Ah, 080h
        DB  0F3h, 0F3h, 00Ch, 00Ch, 03Fh, 0FFh, 080h, 000h, 0F0h, 00Ch
        DB  00Ah, 0A2h, 03Ch, 0CFh, 083h, 030h, 03Fh, 0FFh, 080h, 000h
        DB  0F0h, 008h, 00Ah, 0A8h, 0C0h, 03Fh, 02Ah, 0C0h, 03Fh, 0FFh
        DB  080h, 000h, 0C0h, 00Ch, 02Ah, 0A2h, 03Ch, 0CFh, 083h, 030h
        DB  000h, 000h, 0AAh, 0AAh, 000h, 03Fh, 0AAh, 080h, 0F3h, 0F3h
        DB  00Ch, 00Ch, 0C0h, 000h, 02Ah, 0AAh, 000h, 0FFh, 0AAh, 000h
        DB  0FFh, 0FFh, 000h, 000h, 0F0h, 000h, 00Ah, 0AAh, 003h, 0FFh
        DB  0A8h, 000h, 0FFh, 0FFh, 000h, 000h, 03Ch, 0FFh, 0C3h, 000h
        DB  0CFh, 03Fh, 030h, 0C0h, 0FFh, 0FFh, 000h, 000h, 0C0h, 000h
        DB  03Fh, 0FFh, 000h, 0FFh, 0FFh, 000h, 0FFh, 0FFh, 000h, 000h
rcopter_right0  label   byte
        DB  00Eh, 006h, 0FFh, 0FFh, 000h, 000h, 0F0h, 000h, 00Fh, 0FFh
        DB  000h, 000h, 0FFh, 0FFh, 0FFh, 0FFh, 000h, 000h, 0FFh, 0FFh
        DB  000h, 000h, 000h, 000h, 0FFh, 0FFh, 0FFh, 0FFh, 000h, 000h
        DB  0FFh, 0FFh, 000h, 000h, 00Fh, 0FFh, 0F0h, 000h, 0FFh, 0FFh
        DB  080h, 000h, 0FFh, 0C0h, 000h, 02Ah, 000h, 03Fh, 0AAh, 080h
        DB  0FCh, 0FFh, 003h, 000h, 0FFh, 003h, 000h, 0A8h, 0FFh, 0CFh
        DB  000h, 020h, 0FCh, 0FFh, 003h, 000h, 0FCh, 00Fh, 002h, 0A0h
        DB  0FFh, 0F3h, 000h, 008h, 0FCh, 0FCh, 003h, 002h, 030h, 00Fh
        DB  08Ah, 0A0h, 0FFh, 0FCh, 000h, 002h, 000h, 003h, 0FFh, 0FCh
        DB  0C0h, 00Fh, 02Ah, 0A0h, 0FFh, 0FCh, 000h, 002h, 0FCh, 0FCh
        DB  003h, 002h, 030h, 003h, 08Ah, 0A8h, 0FFh, 0FCh, 000h, 002h
        DB  0FCh, 0FFh, 003h, 000h, 0FCh, 000h, 002h, 0AAh, 000h, 000h
        DB  0AAh, 0AAh, 0FCh, 0FFh, 003h, 000h, 0FFh, 000h, 000h, 0AAh
        DB  000h, 003h, 0AAh, 0A8h, 0FFh, 0FFh, 000h, 000h, 0FFh, 0C0h
        DB  000h, 02Ah, 000h, 0FFh, 0AAh, 0A0h, 0FFh, 0FFh, 000h, 000h
        DB  0FCh, 0F3h, 003h, 00Ch, 0FFh, 03Ch, 000h, 0C3h, 0FFh, 0FFh
        DB  000h, 000h, 0FFh, 000h, 000h, 0FFh, 000h, 003h, 0FFh, 0FCh
rcopter_right1  label   byte
        DB  00Eh, 006h, 0FFh, 0FFh, 000h, 000h, 0F0h, 000h, 00Fh, 0FFh
        DB  000h, 000h, 0FFh, 0FFh, 0FFh, 0FFh, 000h, 000h, 0F0h, 000h
        DB  00Fh, 0FFh, 00Fh, 0FFh, 0F0h, 000h, 0FFh, 0FFh, 000h, 000h
        DB  0FFh, 0FFh, 000h, 000h, 00Fh, 0FFh, 0F0h, 000h, 0FFh, 0FFh
        DB  000h, 000h, 0FFh, 0C0h, 000h, 02Ah, 000h, 03Fh, 0AAh, 080h
        DB  0FFh, 0FFh, 000h, 000h, 0FFh, 003h, 000h, 0A8h, 0FFh, 0CFh
        DB  000h, 020h, 0CFh, 0CFh, 030h, 030h, 0FCh, 00Fh, 002h, 0A0h
        DB  0FFh, 0F3h, 000h, 008h, 0F3h, 03Ch, 00Ch, 0C2h, 030h, 00Fh
        DB  08Ah, 0A0h, 0FFh, 0FCh, 000h, 002h, 0FCh, 003h, 003h, 0A8h
        DB  0C0h, 00Fh, 02Ah, 0A0h, 0FFh, 0FCh, 000h, 002h, 0F3h, 03Ch
        DB  00Ch, 0C2h, 030h, 003h, 08Ah, 0A8h, 0FFh, 0FCh, 000h, 002h
        DB  0CFh, 0CFh, 030h, 030h, 0FCh, 000h, 002h, 0AAh, 000h, 000h
        DB  0AAh, 0AAh, 0FFh, 0FFh, 000h, 000h, 0FFh, 000h, 000h, 0AAh
        DB  000h, 003h, 0AAh, 0A8h, 0FFh, 0FFh, 000h, 000h, 0FFh, 0C0h
        DB  000h, 02Ah, 000h, 00Fh, 0AAh, 0A0h, 0FFh, 0FFh, 000h, 000h
        DB  0FCh, 0F3h, 003h, 00Ch, 0FFh, 03Ch, 000h, 0C3h, 0FFh, 0FFh
        DB  000h, 000h, 0FFh, 000h, 000h, 0FFh, 000h, 003h, 0FFh, 0FCh
bcopter_left0   label   byte
        DB  00Eh, 006h, 000h, 000h, 0FFh, 0FFh, 000h, 00Fh, 0FFh, 0F0h
        DB  0FFh, 0FFh, 000h, 000h, 000h, 000h, 0FFh, 0FFh, 0FFh, 0FFh
        DB  000h, 000h, 0FFh, 0FFh, 000h, 000h, 0FFh, 0F0h, 000h, 00Fh
        DB  0FFh, 0FFh, 000h, 000h, 0FFh, 0FFh, 000h, 000h, 0FCh, 000h
        DB  001h, 055h, 003h, 0FFh, 054h, 000h, 0FFh, 0FFh, 000h, 000h
        DB  0F3h, 0FFh, 004h, 000h, 0C0h, 0FFh, 015h, 000h, 0FFh, 0FFh
        DB  000h, 000h, 0CFh, 0FFh, 010h, 000h, 0F0h, 03Fh, 005h, 040h
        DB  0F3h, 0F3h, 00Ch, 00Ch, 03Fh, 0FFh, 040h, 000h, 0F0h, 00Ch
        DB  005h, 051h, 03Ch, 0CFh, 043h, 030h, 03Fh, 0FFh, 040h, 000h
        DB  0F0h, 003h, 005h, 054h, 0C0h, 03Fh, 015h, 0C0h, 03Fh, 0FFh
        DB  040h, 000h, 0C0h, 00Ch, 015h, 051h, 03Ch, 0CFh, 043h, 030h
        DB  000h, 000h, 055h, 055h, 000h, 03Fh, 055h, 040h, 0F3h, 0F3h
        DB  00Ch, 00Ch, 0C0h, 000h, 015h, 055h, 000h, 0FFh, 055h, 000h
        DB  0FFh, 0FFh, 000h, 000h, 0F0h, 000h, 005h, 055h, 003h, 0FFh
        DB  054h, 000h, 0FFh, 0FFh, 000h, 000h, 03Ch, 0FFh, 0C3h, 000h
        DB  0CFh, 03Fh, 030h, 0C0h, 0FFh, 0FFh, 000h, 000h, 0C0h, 000h
        DB  03Fh, 0FFh, 000h, 0FFh, 0FFh, 000h, 0FFh, 0FFh, 000h, 000h
bcopter_left1   label   byte
        DB  00Eh, 006h, 000h, 000h, 0FFh, 0FFh, 000h, 00Fh, 0FFh, 0F0h
        DB  0FFh, 0FFh, 000h, 000h, 0FFh, 0F0h, 000h, 00Fh, 000h, 00Fh
        DB  0FFh, 0F0h, 0FFh, 0FFh, 000h, 000h, 0FFh, 0F0h, 000h, 00Fh
        DB  0FFh, 0FFh, 000h, 000h, 0FFh, 0FFh, 000h, 000h, 0FCh, 000h
        DB  001h, 055h, 003h, 0FFh, 054h, 000h, 0FFh, 0FFh, 000h, 000h
        DB  0F3h, 0FFh, 004h, 000h, 0C0h, 0FFh, 015h, 000h, 0FFh, 03Fh
        DB  000h, 0C0h, 0CFh, 0FFh, 010h, 000h, 0F0h, 03Fh, 005h, 040h
        DB  0FFh, 03Fh, 000h, 0C0h, 03Fh, 0FFh, 040h, 000h, 0F0h, 00Ch
        DB  005h, 051h, 03Fh, 03Fh, 040h, 0C0h, 03Fh, 0FFh, 040h, 000h
        DB  0F0h, 003h, 005h, 054h, 0C0h, 000h, 03Fh, 0FFh, 03Fh, 0FFh
        DB  040h, 000h, 0C0h, 00Ch, 015h, 051h, 03Fh, 03Fh, 040h, 0C0h
        DB  000h, 000h, 055h, 055h, 000h, 03Fh, 055h, 040h, 0FFh, 03Fh
        DB  000h, 0C0h, 0C0h, 000h, 015h, 055h, 000h, 0FFh, 055h, 000h
        DB  0FFh, 03Fh, 000h, 0C0h, 0F0h, 000h, 005h, 055h, 003h, 0FFh
        DB  054h, 000h, 0FFh, 0FFh, 000h, 000h, 03Ch, 0FFh, 0C3h, 000h
        DB. 0CFh, 03Fh, 030h, 0C0h, 0FFh, 0FFh, 000h, 000h, 0C0h, 000h
        DB  03Fh, 0FFh, 000h, 0FFh, 0FFh, 000h, 0FFh, 0FFh, 000h, 000h
bcopter_right0  label   byte
        DB  00Eh, 006h, 0FFh, 0FFh, 000h, 000h, 0F0h, 000h, 00Fh, 0FFh
        DB  000h, 000h, 0FFh, 0FFh, 0FFh, 0FFh, 000h, 000h, 0FFh, 0FFh
        DB  000h, 000h, 000h, 000h, 0FFh, 0FFh, 0FFh, 0FFh, 000h, 000h
        DB  0FFh, 0FFh, 000h, 000h, 00Fh, 0FFh, 0F0h, 000h, 0FFh, 0FFh
        DB  000h, 000h, 0FFh, 0C0h, 000h, 015h, 000h, 03Fh, 055h, 040h
        DB  0FCh, 0FFh, 003h, 000h, 0FFh, 003h, 000h, 054h, 0FFh, 0CFh
        DB  000h, 010h, 0FCh, 0FFh, 003h, 000h, 0FCh, 00Fh, 001h, 050h
        DB  0FFh, 0F3h, 000h, 004h, 0FCh, 0FCh, 003h, 001h, 030h, 00Fh
        DB  045h, 050h, 0FFh, 0FCh, 000h, 001h, 000h, 003h, 0FFh, 0FCh
        DB  0C0h, 00Fh, 015h, 050h, 0FFh, 0FCh, 000h, 001h, 0FCh, 0FCh
        DB  003h, 001h, 030h, 003h, 045h, 054h, 0FFh, 0FCh, 000h, 001h
        DB  0FCh, 0FFh, 003h, 000h, 0FCh, 000h, 001h, 055h, 000h, 000h
        DB  055h, 055h, 0FCh, 0FFh, 003h, 000h, 0FFh, 000h, 000h, 055h
        DB  000h, 003h, 055h, 054h, 0FFh, 0FFh, 000h, 000h, 0FFh, 0C0h
        DB  000h, 015h, 000h, 00Fh, 055h, 050h, 0FFh, 0FFh, 000h, 000h
        DB  0FCh, 0F3h, 003h, 00Ch, 0FFh, 03Ch, 000h, 0C3h, 0FFh, 0FFh
        DB  000h, 000h, 0FFh, 000h, 000h, 0FFh, 000h, 003h, 0FFh, 0FCh
bcopter_right1  label   byte
        DB  00Eh, 006h, 0FFh, 0FFh, 000h, 000h, 0F0h, 000h, 00Fh, 0FFh
        DB  000h, 000h, 0FFh, 0FFh, 0FFh, 0FFh, 000h, 000h, 0F0h, 000h
        DB  00Fh, 0FFh, 00Fh, 0FFh, 0F0h, 000h, 0FFh, 0FFh, 000h, 000h
        DB  0FFh, 0FFh, 000h, 000h, 00Fh, 0FFh, 0F0h, 000h, 0FFh, 0FFh
        DB  000h, 000h, 0FFh, 0C0h, 000h, 015h, 000h, 03Fh, 055h, 040h
        DB  0FFh, 0FFh, 000h, 000h, 0FFh, 003h, 000h, 054h, 0FFh, 0CFh
        DB  000h, 010h, 0CFh, 0CFh, 030h, 030h, 0FCh, 00Fh, 001h, 050h
        DB  0FFh, 0F3h, 000h, 004h, 0F3h, 03Ch, 00Ch, 0C1h, 030h, 00Fh
        DB  045h, 050h, 0FFh, 0FCh, 000h, 001h, 0FCh, 003h, 003h, 054h
        DB  0C0h, 00Fh, 015h, 050h, 0FFh, 0FCh, 000h, 001h, 0F3h, 03Ch
        DB  00Ch, 0C1h, 030h, 003h, 045h, 054h, 0FFh, 0FCh, 000h, 001h
        DB  0CFh, 0CFh, 030h, 030h, 0FCh, 000h, 001h, 055h, 000h, 000h
        DB  055h, 055h, 0FFh, 0FFh, 000h, 000h, 0FFh, 000h, 000h, 055h
        DB  000h, 003h, 055h, 054h, 0FFh, 0FFh, 000h, 000h, 0FFh, 0C0h
        DB  000h, 015h, 000h, 00Fh, 055h, 050h, 0FFh, 0FFh, 000h, 000h
        DB  0FCh, 0F3h, 003h, 00Ch, 0FFh, 03Ch, 000h, 0C3h, 0FFh, 0FFh
        DB  000h, 000h, 0FFh, 000h, 000h, 0FFh, 000h, 003h, 0FFh, 0FCh
balloon label   byte
        DB  014h, 006h, 0FFh, 0C0h, 000h, 03Fh, 003h, 0FFh, 0FCh, 000h
        DB  0FCh, 000h, 003h, 0EAh, 000h, 03Fh, 057h, 0C0h, 0F0h, 000h
        DB  00Dh, 06Ah, 000h, 00Fh, 056h, 0B0h, 0C0h, 000h, 035h, 06Ah
        DB  000h, 003h, 056h, 0ACh, 0C0h, 000h, 035h, 0AAh, 000h, 000h
        DB  055h, 0ABh, 000h, 000h, 0D5h, 0AAh, 000h, 000h, 055h, 0ABh
        DB  000h, 000h, 0D5h, 0AAh, 000h, 000h, 055h, 0ABh, 000h, 000h
        DB  0D5h, 0AAh, 000h, 000h, 055h, 0ABh, 0C0h, 000h, 035h, 0AAh
        DB  000h, 003h, 055h, 0ACh, 0C0h, 000h, 035h, 06Ah, 000h, 003h
        DB  056h, 0ACh, 0C0h, 000h, 035h, 06Ah, 000h, 003h, 056h, 0ACh
        DB  0F0h, 000h, 00Fh, 0EAh, 000h, 00Fh, 057h, 0F0h, 0F3h, 0C0h
        DB  00Ch, 03Fh, 003h, 0CFh, 0FCh, 030h, 0FCh, 0FFh, 003h, 000h
        DB  0FFh, 03Fh, 000h, 0C0h, 0FCh, 0FFh, 003h, 000h, 0FFh, 03Fh
        DB  000h, 0C0h, 0FFh, 03Fh, 000h, 0C0h, 0FCh, 0FFh, 003h, 000h
        DB  0FFh, 03Fh, 000h, 0C0h, 0FCh, 0FFh, 003h, 000h, 0FFh, 0C0h
        DB  000h, 03Fh, 003h, 0FFh, 0FCh, 000h, 0FFh, 0C0h, 000h, 03Fh
        DB  003h, 0FFh, 0FCh, 000h, 0FFh, 0F0h, 000h, 00Fh, 00Fh, 0FFh
        DB  0F0h, 000h
;
;Below is a form for a crescent moon in a form suitable for a BYTE MOVE
; driver. The format is:
;       byte 0 - height in lines (h)
;       byte 1 - width in bytes  (w)
;       followed by w X h image bytes
;
moon    label   byte
```


Make Flashy Programs.

In dBASE, Turbo Pascal, BASIC, C, COBOL, Fortran, 1-2-3, DOS.

New! Flash-Up™ Windows.



Instantly flash-up menus and help windows.

From within almost any program language or application program. When the user makes a menu selection, a sequence of keystrokes is sent to the running program as if the user were typing it at the keyboard.

A keyboard macro utility that your programs can control!

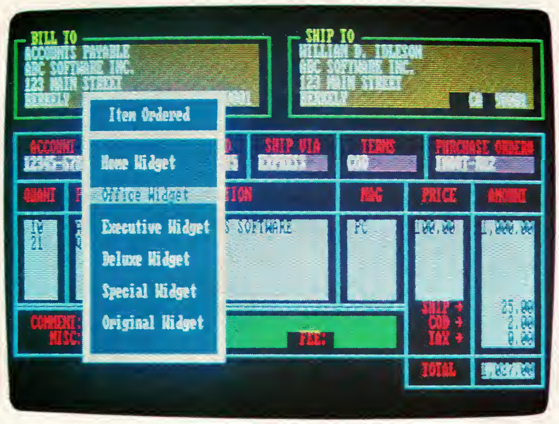
Just like keyboard macro utilities, Flash-Up Windows can be used to send repetitive keystrokes and simplify commands for your users.

But Flash-Up Windows can do more. Your programs can send commands directly to Flash-Up. Powerful commands that control when and how windows appear. Flash-Up is a perfect tool for controlling programs and entire systems via menus.

The powerful window editor allows you to create windows, change their size, location, and color — all automatically.

Works with DOS, BASIC, Turbo Pascal, C, COBOL, Fortran, dBASE, 1-2-3, R:BASE ...and most other software. $90.

Flash Code.™ For dBASE II, III and III+.



The most powerful, most complete program developer for dBASE II, III, and III+.

Use Flash Code's advanced screen design editor and see how incredibly easy screen design can be. Draw boxes, lines, and select colors. Specify special parameters for edit checking, initial values, calculated fields, etc. All within one, easy to use editor.

Then tell Flash Code to write the program. Based on your screen designs, Flash Code can generate a full database program. Well written code that you can use as is, or merge with your own programs.

Menus and help windows add a new dimension to your dBASE applications.

Flash Code includes its own dBASE window editor. Use the window module to flash-up windows and screens instantly. The windows are easily converted when you move to the added power of our Flash-Up Windows product.

"... a truly remarkable product. Brand new, state-of-the-art and actually fun to use. The windows make it an even more amazing product."
— Gary N. Prague, Author, "The dBase III Programming Handbook"

Flash Code. $150.

Screen Sculptor.™ For IBM BASIC, Turbo Pascal, and Quick Basic.

Create screens in minutes, then Screen Sculptor writes the program.

In IBM BASIC, Turbo and Microsoft Pascal, and Quick Basic. Simply "draw" your screens with our advanced screen design editor and replace hours of tedious work with minutes of creative design.

Move pieces of the screen around, select colors from a menu, draw boxes, lines, paint, repeat last character. And more! Specify variable names, data types, acceptable data ranges, pictures for edit checking, etc., and you're ready to go.

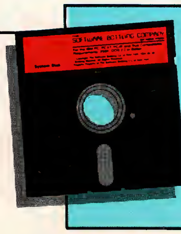
Generate tested program source code to merge with your own program.

Based on your screen design. The generated program flashes up the screen, allows data to be entered by the user, and edit checks the input data.

"... so well done that you may not need to refer to the manual."

"... an exceptional product that fills a real need in the development of new programs."
— Computer Language

Screen Sculptor. $125.



No Risk Demo Offer!

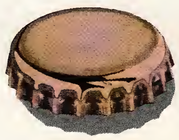
Order any package and receive a separately sealed demo disk of the product. Use the demo and the manual for 30 days. If you're not satisfied — for any reason — return the entire package for a full refund.

Credit card orders call 24 hrs/day: 1(800) 824-7888, operator 268.

All other orders and inquiries call or write Software Bottling Co., 6600 L.I. Expwy., Maspeth, NY 11378. 718-458-3700.

Requires an IBM PC, XT, AT or true compatible, 256K.

Not copy protected.



THE SOFTWARE BOTTLING COMPANY OF NEW YORK

For $295 you can draw your own conclusions.

Drafix™ 1. The first high performance CAD software everyone can afford.

Low-cost, high performance computer-aided design & drafting (CAD) software is finally available for use on your PC.

There's never been a professional CAD package so complete at such a low cost.

Drafix 1 from Foresight Resources. Only $295.

Powerful, full-featured second generation CAD software.

Drafix 1 offers all of the capabilities you expect to find in packages costing $2,000 or more.

This is not simply a souped-up paint package. Drafix 1 is a breakthrough in design that organizes sophisticated CAD functions into smooth, fast operations. All of the drawing, designing and editing functions that designers demand from a serious CAD tool are included in Drafix 1.

A breakthrough in screen design and visual user interface.

Drafix 1 features the best organized screen design ever devised. Very simply, it displays all the information you need, all of the time.

The entire menu hierarchy is displayed constantly. There's no need to memorize commands or search for menus.

Snap-mode options are continually shown on the left screen border and can be selected "on the fly" by either pointer device or a single keystroke. Roll-down screens provide quick access to the virtually unlimited drawing, display and editing options.

And it's all controlled by a versatile three-button mouse, or digitizer, with on-screen prompts so each button function is clearly defined.

AutoCAD® compatible for easy expansion.

If you or someone in your company already uses AutoCAD, Drafix 1 offers an inexpensive way to add to your capability.

The optional Drafix 1/AutoCAD file exchange utility permits transfer of drawings between Drafix 1 and AutoCAD.

It's the perfect, low-cost alternative for increasing your drafting and design capacity.

Unbelievable low bundle prices.

To get you up and running we offer two special hardware bundles — all the equipment you need at prices you won't believe.

Get Drafix 1 with your choice of Torrington's Manager Mouse or the LOGIMOUSE® C7 for just $395.00. Or get Drafix 1 with the SummaSketch 12″ × 12″ digitizer tablet with stylus for just $650.00.

Order today. Take advantage of our 30-day, money back guarantee.

We're so certain you'll like the performance and versatility of Drafix 1 you're welcome to try it risk free for 30 days. If you're not completely satisfied, return it to us for a full refund.

Find out just how good CAD software can be. For only $295.00.

Use our toll-free number and any major credit card to order your copy of Drafix 1 today.

SPECIFICATIONS

Code specifications
- Lattice "C" language
- Halo graphics
- Floating point database
- Coprocesser support
- PC-DOS/MS-DOS 2.1 or later

Drafix 1 Features

Items
- Lines
- Pt markers
- Arcs & circles
- Note text
- Polygons & ellipses
- Symbols/individual & nested

Item attributes
- 16 pen colors
- 255 Layers
- 8 Linetypes
- 12 Text fonts
- 32 Pt marker types

Screen Display
- Zoom/Pan/Full
- 8 Save Views
- Slide save/view
- Grids on/off

Metric & English Standards
- Engineering (decimal/fractional)
- Architectural (ft in)

Project drawing info.

Numeric input
- Keyboard and cursor
- Absolute. Relative. Polar

Snap Modes
- Gridpoint
- Endpoint
- Midpoint
- Intersect
- On item
- Quadrant
- Tangent
- Arc center

Transform/Copy
- Move
- Rotate
- Scale
- Mirror
- Align

Modify/Edit
- Attributes
- Break/Divide
- Fillet/chamfer
- Trim
- Stretch
- Erase
- Explode symbols & polygons
- Mask
- Replace
- Merge
- Region select
- Workgroup

Auto hatching, polygon fill

Auto Dimensions
- Linear Hor/Ver/Aligned
- Angular
- Dia/Radius
- Leader Notes
- Ordinate
- Chain & Baseline

Check calculations
- Coordinates
- Distance & angle
- Area & perimeter

Item Masking

Hardware

Computers
- IBM PC XT AT and compatibles
- AT&T 6300 Plus
- Zenith
- System Requirements
  - 512K RAM
  - RS-232 Comm port
  - Mouse or digitizer
  - Coprocessor recommended

Graphics Display boards
- IBM CGA/EGA and compatibles
- Hercules, and a wide range of graphics boards

Digitizers & Locators
- All popular locator devices supported

Plotters
- All popular plotters ("C" size) supported
- "E" size plotter option available

Printers
- All popular laser and dot matrix printers supported

Order Now! Call Toll-Free!

1-800-231-8574, Ext. 500

☐ Drafix I CAD Package	$295
☐ Drafix I with Mouse Torrington ____ LOGIMOUSE ____	$395
☐ Drafix I with SummaSketch tablet	$650
☐ AutoCAD File Exchange Utility	$ 95

Dealer inquiries welcome

Check, money order, Visa and MasterCard accepted

FORESIGHT RESOURCES CORP.™
932 Massachusetts
Lawrence, KS 66044
913/841-1121

CIRCLE NO. 138 ON READER SERVICE CARD

```
        DB      014h, 004h, 000h, 000h, 03Fh, 0FCh
        DB      000h, 00Fh, 0FFh, 0FFh, 000h, 0FFh, 0FFh, 000h
        DB      003h, 0FFh, 0F0h, 000h, 00Fh, 0FFh, 0C0h, 000h
        DB      03Fh, 0FFh, 000h, 000h, 03Fh, 0FFh, 000h, 000h
        DB      0FFh, 0FFh, 000h, 000h, 0FFh, 0FCh, 000h, 000h
        DB      0FFh, 0FCh, 000h, 000h, 0FFh, 0FCh, 000h, 000h
        DB      0FFh, 0FCh, 000h, 000h, 0FFh, 0FFh, 000h, 000h
        DB      03Fh, 0FFh, 000h, 000h, 03Fh, 0FFh, 000h, 000h
        DB      00Fh, 0FFh, 0C0h, 000h, 003h, 0FFh, 0F0h, 000h
        DB      000h, 0FFh, 0FFh, 000h, 000h, 00Fh, 0FFh, 0FFh
        DB      000h, 000h, 03Fh, 0FCh, 000h, 000h, 000h, 000h
        DB      03Fh, 0FFh, 000h, 000h, 000h, 000h, 00Fh, 0FFh
        DB      0C0h, 000h, 000h, 000h, 003h, 0FFh, 0F0h, 000h
        DB      000h, 000h, 000h, 0FFh, 0FFh, 000h, 000h, 000h
        DB      000h, 00Fh, 0FFh, 0FFh, 0C0h, 000h, 000h, 000h
;
;The tables below define the x,y and color of pixels which represent
; stars in a night sky.
;
starx   label   byte
        db     10, 50, 39, 22, 41,  2, 50, 33, 65, 41, 15, 19, 19, 54
        db     34, 61, 19, 41, 22, 55, 23, 68,  8, 29, 51, 71, 39, 30
        db     41,  9, 21, 13, 68, 61, 43, 70, 40, 17, 39, 19, 29, 45
num_stars       equ     $-starx
stary   label   byte
        db      7, 33, 90,100, 80, 35, 92,172,106, 97,123,168, 42, 88
        db    150, 34, 29, 90,170, 65, 39, 99,145,163,109, 78,127,166
        db    169, 84, 15, 65, 99,155, 89, 58, 25, 80, 76,168, 98, 99
star_color label   byte
        db      1,  2,  3,  1,  2,  3,  1,  2,  3,  1,  2,  3,  1,  2
        db      3,  1,  2,  3,  1,  2,  3,  1,  2,  3,  1,  2,  3,  1
        db      2,  3,  1,  2,  3,  1,  2,  3,  1,  2,  3,  1,  2,  3
;
start   proc    near
        cld                             ;count up
        push    cs                      ;most data is kept in code segment
        pop     ds                      ;
;set 320x200x4 graphics mode
        mov     ax,0004h                ;graphics mode 4 is supported by both
        int     10h                     ; the EGA and CGA
;select palette 0
        mov     ah,0bh                  ;set color palette function
        mov     bh,1                    ;select palette to be used subfunction
        mov     bl,0                    ;select palette 0 (green/red/yellow)
        int     10h                     ;
;put stars into background buffer
        mov     ax,background_segment
        mov     es,ax
        mov     bx,num_stars-1
        mov     dl,80                   ;lines are 80 bytes long
next_star:
        sub     di,di                   ;calculate star's screen buffer
        mov     al,[stary+bx]           ; address based on line number
        shr     al,1                    ; and byte position from tables
        jnc     calc_y_offset           ;
        mov     di,2000h                ;odd lines in second bank
calc_y_offset:
        mul     dl                      ;number of even lines * 80
        mov     cl,[starx+bx]           ; plus byte position in line
        sub     ch,ch                   ; plus odd line offset
        add     ax,cx                   ; is screen buffer offset
        add     di,ax                   ;
        mov     al,[star_color+bx]      ;get the star image byte (all are
        or      es:[di],al              ; single pixels) and OR it into
        dec     bx                      ; the background buffer.
        jns     next_star               ;
;put moon into background buffer
        mov     ax,background_segment
        mov     es,ax                   ;put forms into background buffer
        mov     bx,88                   ;put moon image on line 88
        mov     cx,37                   ; and byte 37
        mov     si,offset moon          ;
        call    byte_move_form_driver
;put horizon into background buffer
        mov     ax,background_segment   ;horizon is two consecutive
        mov     es,ax                   ; red lines
        mov     ax,0aaaah               ;8 red pixels
        mov     di,92*80                ;point to scan line 184
        mov     cx,40                   ;width of screen in words
        rep stosw
        mov     di,92*80+2000h          ;point to scan line 185
        mov     cx,40                   ; and draw it
        rep stosw
;copy background to display memory
        mov     ax,screen_buffer_adr    ;
        mov     es,ax                   ;
        mov     ax,background_segment   ;
        mov     ds,ax                   ;
        sub     si,si                   ;
        sub     di,di                   ;
        mov     cx,2000h                ;
        rep movsw                       ;
;
;Initialize the object services routine
;
        call    initialize
;
;Orchestrate the objects' movements
;
        push    cs                      ;object data is kept in
        pop     ds                      ; code segment
iteration_loop:                         ;sequence all objects once
        mov     di,((num_objects-1) * 2) ;start with last object
next_object:
;see if it's time to manipulate object
        dec     [delay+di]              ;each delay count is 1/60 second
        jnz     done_with_object        ; so long as program can keep up
;see if it's time to get next set of control variables
        dec     [repeat+di]             ;each count is the number of
        jnz     reset_delay             ; movements using the present set
                                        ; of control values
;get next set of control parameters
        mov     si,[control+di]         ;get the addr of the parameter list
        lodsw                           ;get the number of bytes to move
        mov     [xinc+di],ax            ; horizontally each time delay
                                        ; counts down to zero
        lodsw                           ;get the number of lines to move
        mov     [yinc+di],ax            ; vertically each count down
        lodsw                           ;get pointer to the table of
        mov     [form+di],ax            ; form addres to be sequenced though
        mov     [formbase+di],ax        ;set to repeat with starting form
        lodsw                           ; when the end of the table is hit
        mov     [formmax+di],ax         ;save end of table marker
        lodsw                           ;get next delay count which controls
        mov     [delaybase+di],ax       ; the speed
        lodsw                           ;get the number of times to use
        mov     [repeat+di],ax          ; this new set of control parameters
        cmp     [controlmax+di],si;point to next list of control
        ja      save_control_base       ; parameters to be used.
        mov     si,[controlbase+di]     ; reycle when end of list hit
save_control_base:
        mov     [control+di],si         ;next control list entry address
;
reset_delay:
        mov     ax,[delaybase+di]       ;reset iteration count until
        mov     [delay+di],ax           ; next manipulation of this object
;find object's new position
        mov     cx,[x+di]               ;calculate new x and y positions
        add     cx,[xinc+di]            ; and store them
        mov     [x+di],cx               ;
        mov     bx,[y+di]               ;
        add     bx,[yinc+di]            ;
        mov     [y+di],bx               ;
;point to new form for object so that it appears to move internally
        mov     si,[form+di]            ;each time an object is counted down
        inc     si                      ; the form used is taken from the
        inc     si                      ; next address in table
        cmp     [formmax+di],si         ;
        ja      setform                 ;
        mov     si,[formbase+di]        ;
setform:
        mov     [form+di],si            ;cycle to the next form for the object
        mov     si,[si]                 ;put pointer to object's form in SI
        shr     di,1                    ;object number in DI
                                        ;BX holds line number, CX byte number
        call    object_services         ;register change in object's status
        shl     di,1                    ;turn object number into object index
;point to next object
done_with_object:
        dec     di
        dec     di
        js      check_for_keystroke     ;see if finished with iteration
```

```
        jmp     next_object          ; if not process next object
;
;Check for keystroke; if so, clear key & end.
;
check_for_keystroke:
        mov     ah,1                    ;see if a char is waiting
        int     16h                     ;get keyboard status
        jnz     getkey                  ;if zero no key waiting
;
;Wait for new vertical sync (CGA) or retrace (EGA) before putting
; objects on screen to avoid undesirable screen breakup and blinking
; effects. Waiting also keeps the objects moving a constant rate.
;
if cga
        mov     dx,3dah               ;point to cga status register
wait_for_not_vert_sync:
        in      al,dx                 ;wait for begining of vertical
        and     al,08h                ; sunc by starting search from
        jnz     wait_for_not_vert_sync ; outside of sync pulse
wait_for_vert_sync:
        in      al,dx                 ;wait for start of sync
        and     al,08h                ;
        jz      wait_for_vert_sync    ;
;
        pushf                         ;push flags to simulate an
        call    put_objects_on_screen ; interrupt
endif
if ega
        mov     ax,[old_vert_count]        ;synchronize animation loop
wait_for_after_interrupt:                  ; to vert interrupt so it
        cmp     ax,[vert_int_modulo_count] ; will run at a fixed
        je      wait_for_after_interrupt   ; smooth rate
        mov     ax,[vert_int_modulo_count] ;save new as old for next
        mov     [old_vert_count],ax        ; time
endif
        jmp     iteration_loop             ;

;
;Run is completed.
;
getkey:
        sub     ah,ah      ;get key that was hit
        int     16h        ;
done:
        call    terminate  ;do any clean up required by the sprite
;                          ; routines
;Set to text mode
;
        mov     ax,0002h   ;set the adapter to video mode 2
        int     10h        ;
;
;Return to DOS
;
        mov     ax,4c00h   ;terminate program with return code of zero
        int     21h        ;
start   endp
cseg    ends
        end     start      ;program execution will start at start
```

LISTING 2: SPRITES2.ASM

```
;
;These routines produce the effect of hardware sprites in software
; on IBM PC compatible computers. They put objects onto the screen
; in a manner which preserves the background, and produces no
; undesirable fringe or overlap effects. Operations which affect
; video buffer memory are performed as much as possible during
; video non-display periods to avoid other undesirable effects.
;
; Entry points and parameters:
;
;   Initialize - Sets the background buffer address to be used to erase
;       objects, resets internal flags and queue, and on EGAs
;       sets up the use of the vertical interrupt to drive the
;       drawing routines.
;
;       Inputs  - AX holds paragraph address of background buffer.
;       Outputs - None.
;
;   Terminate  - Resets the EGA vertical interrupt hardware and vector
;
;       Inputs  - None.
;       Outputs - None.
;
;   Object_services - Sets X,Y and Form address for a given object
;       to be drawn, and activates or deactivates the object.
;
;       Inputs  - CX holds X position in bytes (0-79) of upper
;                 left hand corner of object. 0 is leftmost.
;               - BX holds Y position in lines (0-198) 0 is top.
;                 BX must be even! Objects cannot start on odd lines.
;                 Objects must also be an even number of lines high.
;               - DI holds object number. Higher numbered objects
;                 will appear to be in front of lower numbered
;                 objects when they overlap.
;               - SI holds the offset in the code segment of the form
;                 to be drawn for the object. A value of Offffh means
;                 that the object is to be erased, then ignored.
;                 Forms must be in the following format:
;
;                    byte 0 - height in lines (h)
;                    byte 1 - width in bytes  (w)
;                    followed by w X h (mask word, image word) pairs.
;
;       Outputs - None.
;
;       Registers - All are saved, except flags
;
;       Warning - No bounds checking is done. X,Y or object numbers
;                 out of range can send your program into hyperspace.
;
;   Put_objects_on_screen - This routine should be called by a program
;       running on a CGA to put the objects on the screen. It must be
;       far-called as if it were an interrupt routine. For example:
;
;                      pushf
;                      call far ptr put_objects_on_screen
;
;       For best results this routine should be called immediately
;        upon the sensing of vertical retrace.
;
;       Inputs  - None.
;       Outputs - None.
;
; Vert_int_modulo_count - This memory word is incremented each time
;   the objects are put into the screen map. On EGAs it can be used
;   to synchronize a program to the constant time base provided by
;   the vertical interrupt.
;
;The flag below must be set properly before assembling this program
;
ega     equ  0           ;1 to assemble for Enhanced Graphics Adapter
                         ;0 to assemble for Color Graphics Adapter
cga     equ  (ega xor 1) ;the opposite status of ega
;
bios_data_segment segment at 40h       ;BIOS keeps its data at 400h;
        org     63h                    ;at 463h is a word that holds
bios_crtc_base_address  dw      ?      ; the CRT controller's base
bios_data_segment ends                 ; address
;
;
cseg    segment para public 'cseg'
        assume  cs:cseg,ds:cseg,es:nothing
        public  initialize,terminate,object_services
        public  put_objects_on_screen,vert_int_modulo_count
;
;Memory for the parameters used to keep track of objects is reserved
; below. Many of the parameters stored are very code specific so that
; the size and number of objects which could be processed during
; vertical non-display time could be maximized.
;
number_of_objects equ   3  ;this should be set to the maximum number
                           ; of objects or priorities which will
                           ; need to be kept track of at one time.
;
queue label word
;
draw_screen_offset      dw ?  ;offset in screen memory buffer of upper
                              ; left hand corner of object. Offffh if
                              ; object is to be ignored.
dist_to_odd_scan_line   dw ?  ;distance from end of object on an even
```

```
                                ; scan line to the start of the object
                                ; on the next (odd) scan line
dist_to_even_scan_line  dw ?  ;distance from end of object on an odd
                                ; scan line to the start of the object
                                ; on the next (even) scan line
erase_parms label word
erase_width          dw ?  ;the object's screen image width in words
erase_entry_point    dw ?  ;the address of the inline code to do erase
erase_screen_offset  dw ?  ;the address where object was last drawn
                            ; 0ffffh if object is not to be erased.
erase_image_offset   dw ?  ;used to determine if need to erase when
                            ; object is in old position
;
length_of_erase_parms equ $-erase_parms
;
draw_col_entry_point dw ?  ;address of the column code for drawing
draw_row_entry_point dw ?  ;address of the row inline code for drawing
draw_image_offset    dw ?  ;offset in the code segment of the image
;
queue_item_length equ ($ - queue) ;number of bytes for each item
distance_from_entry_point_to_next_item equ $ - erase_entry_point
distance_from_image_to_next_item equ $ - draw_image_offset
;
        db    (  (number_of_objects-1) * queue_item_length ) dup(?)
end_of_queue label word
;
vert_int_modulo_count   dw 0  ;incremented each time a vertical
                              ; interrupt occurs
background_segment      dw ?  ;place to hold the paragraph address
                              ; of the background buffer used to
                              ; erase objects
crtc_base_address       dw ?  ;will hold register address
;
old_int10_offset        dw ?  ;place to store the vector contents
old_int10_segment       dw ?  ; so they can be restored when finished
;
old_int_mask            db ?  ;place to store the mask register's
                              ; contents so it can be restored
;
true     equ     1  ;used for flag values
false    equ     0  ;
;
need_to_draw_something_flag db false ;true if a change needs to be
                                     ; made to any of the objects'
                                     ; screen images
;
screen_buffer_paragraph_adr equ 0b800h
;
;
initialize proc near
        cld                               ;count up
        push    ds                        ;
        mov     cs:[background_segment],ax ;store background adr
        mov     ax,cs                     ;make data segment
        mov     ds,ax                     ; same as code segment
                                          ; since that is where data
        mov     es,ax                     ; used by this routine is
        mov     di,offset queue           ;turn off all objects
        mov     cx,(number_of_objects * queue_item_length)/2
        mov     ax,0ffffh                 ;
        rep stosw                         ;
;
        mov     [need_to_draw_something_flag],false ;nothing to draw
if ega
        sub     ax,ax                          ;swapping interrupt
        mov     ds,ax                          ; vectors with our
        mov     bx,(10*4)                      ; interrupt handler
        mov     ax,offset put_objects_on_screen ;our vertical int
        mov     dx,cs                          ; handler address
        cli                                    ;disable interrupts
        xchg    [bx],ax                        ;offset
        xchg    [bx+2],dx                      ;segment
        mov     cs:[old_int10_offset],ax       ;save old value so we
        mov     cs:[old_int10_segment],dx      ; can restore it upon
;                                              ; termination
        mov     ax,bios_data_segment           ;find the register
        mov     ds,ax                          ; address
        assume  ds:bios_data_segment           ;
        mov     dx,[bios_crtc_base_address]    ;
        mov     cs:[crtc_base_address],dx      ;save it in code seg
        mov     al,11h                         ;select vertical
        out     dx,al                          ; retrace end register
        mov     al,04h                         ; and flip it off
        inc     dx                             ;
        out     dx,al                          ;
        mov     al,14h                         ; then flip it on
        out     dx,al                          ;
;
        in      al,21h                         ;enable IRQ2
        mov     cs:[old_int_mask],al           ; save old value
        and     al,not 4                       ;
        out     21h,al                         ;
;
        sti                                    ;enable interrupts
endif
        pop     ds                             ;restore data segment
        ret
initialize endp
;
terminate       proc    near   ;only needs to be used when assembled
if ega                         ; for use on an EGA
        mov     dx,[crtc_base_address]
        mov     al,11h
        out     dx,al
        inc     dx
        mov     al,24h         ;bit 5 high to disable, bit 4 low to
        out     dx,al          ; clear vertical interrupt
        push    ds
        sub     ax,ax                       ;restore original interrupt
        mov     ds,ax                       ; 10 vector
        mov     bx,(10*4)                   ;
        mov     ax,cs:[old_int10_offset]    ;
        mov     dx,cs:[old_int10_segment]   ;
        cli                                 ;make sure interrupt
        mov     [bx],ax                     ; doesn't occur while
        mov     [bx+2],dx                   ; there is an inconsistant
                                            ; vector/mask
        mov     bl,cs:[old_int_mask]        ;restore IRQ2 mask bit
        and     bl,4                        ; to state it had when
        in      al,21h                      ; Initialize was called
        and     al,not 4                    ;
        or      al,bl                       ;
        out     21h,al                      ;
        sti
        pop     ds
endif
        ret
terminate endp
;
object_services proc near
        cld                    ;
        push    es             ;save the registers used
        push    ds             ;
        push    ax             ;
        push    bx             ;
        push    cx             ;
        push    si             ;
        push    di             ;
;
        mov     ax,cs          ;everything will be in code segment
        mov     es,ax          ;
        mov     ds,ax          ;
;
        shl     di,1           ;multiply object number
        shl     di,1           ; which is in DI by 20 to
        mov     ax,di          ; find object's parameter table
        shl     di,1           ; offset in queue structure
        shl     di,1           ; (NOTE: If a code change alter
        add     di,ax          ;  queue_item_length this code must
;                              ;  be changed!)
        mov     ax,offset queue ;point directly to object's first
        add     di,ax          ; parameter
        mov     ax,[bx+even_line_screen_offset_table]
        add     ax,cx          ;find screen offset of top left corner
if ega
        cli      ;can't allow parameters to be just half
                 ; changed if a vertical interrupt occurs
endif
        cmp     si,0ffffh      ;if object is to be turned off then
        jne     save_position  ; need to store a 0ffffh for the draw
        mov     ax,si          ; screen position
```

```
        stosw                   ;
        jmp short finish_services
save_position:                  ;
        stosw           ;save as first parameter (draw_screen_offset)
        lodsb           ;get the height of the image
        xor     ah,ah   ;make height a word
        mov     bx,ax   ;store height
        lodsb           ;get the width of the image in bytes
        mov     cx,2000h ;calculate amount to add after even scan
        sub     cx,ax   ; lines are drawn to get the address of the
        xchg    ax,cx   ; next scan line, and store it in queue
        stosw           ;
        mov     ax,1fb0h ;calculate amount to subtract after odd scan
        add     ax,cx   ; lines are drawn to get the address of the
        stosw           ; next scan line, and store it in queue
        mov     ax,cx   ;store the width in queue
        shr     ax,1    ; width is stored as number of words
        stosw           ;
        mov     ax,[bx+erase_inline_vector_table-2]
                        ;-2 because there is no 0 lines entry point
        stosw           ;store the place to jump to erase an image
                        ; of this height
        add     di,4    ;skip erase_screen_offset and
                        ; erase_image_offset as these are filled
                        ; in when an object is drawn
        xchg    si,cx   ;swap image offset with width
        mov     ax,[si+column_inline_vector_table-2] ;inline code adr
        stosw           ; driver operates with words, so there is no
                        ; need to divide SI by two to do table lookup
        mov     ax,[bx+row_inline_vector_table-2] ;inline code adr
        stosw           ; which calls column inline code for each row
        mov     [di],cx ;last param to put on queue is image offset
;
finish_services:
        mov     [need_to_draw_something_flag],true ;record change
if ega
        sti             ;all parameters have been put on queue
                        ; so interrupts are safe now
endif
        pop     di      ;restore those registers that were used
        pop     si      ;
        pop     cx      ;
        pop     bx      ;
        pop     ax      ;
        pop     ds      ;
        pop     es      ;
        ret             ;
object_services endp
;
;This table is used to find the offset of an even scan line in the
; memory map of the color graphics adapter in medium resolution mode.
;
even_line_screen_offset_table label word
xx=0
        rept    100     ;there are 100 even lines
        dw      xx*50h  ; each is 50h (80 decimal) long
xx=xx+1
        endm
;
use_old_vector:
        pop     ax      ;restore registers used before
        pop     dx      ; jumping to previous IRQ2 handler
        jmp dword ptr cs:[old_int10_offset]
;
put_objects_on_screen proc far
        push    dx      ;save registers used by EGA code
        push    ax      ;
if ega
                        ;must check if interrupt is being signaled
                        ; by the EGA card. If not, it needs to be
        mov     dx,3c2h ; handled by another routine in the vector
        in      al,dx   ; chain. The PC AT in particular uses
        test    al,80h  ; IRQ2 for multiple devices.
        jz      use_old_vector
endif
        inc     cs:[vert_int_modulo_count]      ;count vert interrupts
        sti                                     ;enable interrupts
        cmp     cs:[need_to_draw_something_flag],true ;anything to do?
        je      process_queue                   ; jmp if there is
if ega                                          ; otherwise do nothing
        cli
        mov     al,20h  ;issue a non_specific EOI (End Of Interrupt)
        out     20h,al  ; so that interrupt controller chip will
                        ; acknowledge future vertical interrupts
        mov     dx,cs:[crtc_base_address] ;re-enable ega card interrupt
        mov     al,11h  ; select vertical retrace end
        out     dx,al   ; register and clear vertical interrupt
        mov     al,04h  ;
        inc     dx      ;
        out     dx,al   ;
        mov     al,14h  ; then enable vertical interrupt
        out     dx,al   ;
endif
        pop     ax      ;restore original values
        pop     dx      ;
        iret
;
skip_this_object:
        add     si,queue_item_length            ;point SI to next item
        cmp     si,offset end_of_queue          ;see if we are done
        je      draw                            ; jmp to draw if we are
        jmp     get_next_objects_screen_adr     ; if not, erase next
;
process_queue:
        push    bx      ;save the rest of the world
        push    cx      ;
        push    bp      ;
        push    si      ;
        push    di      ;
        push    ds      ;
        push    es      ;
;
        push    cs      ;setup environment
        pop     ds      ; data is in code segment
        cld             ; we will count up
        mov     ax,screen_buffer_paragraph_adr
        mov     es,ax   ; ES points to screen memory
;
        mov     si,offset queue ;point to beginning of queue
;
;Erase all the active objects which have old screen positions
; different from their present screen position or have different
; image offsets
;
get_next_objects_screen_adr:
        mov     ax,[si+erase_screen_offset-draw_screen_offset]
                                ; get the screen buffer
                                ; offset of last draw
        cmp     ax,0ffffh       ;if ffffh then object is
        je      skip_this_object ; yet to be drawn
        cmp     ax,[si]         ;if new and old positions and
        jne     erase_this_object ; images are same then skip erase
        mov     di,[si+erase_image_offset-draw_screen_offset]
        cmp     di,[si+draw_image_offset-draw_screen_offset]
        je      skip_this_object ;
erase_this_object:
        inc     si              ;point to next parameter
        inc     si              ;
        mov     di,ax           ;save screen buffer adr
        lodsw                   ;get distance to odd scan line
        mov     bp,ax           ; save in BP for inline code use
        lodsw                   ;get distance to even scan line
        mov     dx,ax           ; save in DX for inline code use
        lodsw                   ;get the width in words for erase
        mov     cx,[si]         ;get address of erase inline code
;
        push    si              ;save position so next object
        push    ds              ; can be found
        mov     ds,[background_segment] ;stuff to erase with
        call    cx              ;erase it!
        pop     ds              ;restore where
        pop     si              ; we left off in queue
        add     si,distance_from_entry_point_to_next_item ;next item
        cmp     si,offset end_of_queue  ;see if we are done
        jne     get_next_objects_screen_adr ; jmp if more to erase
;
;Draw all the active objects by AND/ORing into screen buffer
;
draw:
        mov     si,offset queue ;point to beginning of queue
;
get_next_objects_screen_adr2:
```

The Full-Service Source For Programming Software

At Lifeboat, we're committed to more than selling you the best software. We provide tech support for every program we sell. Our expert staff is ready to take your calls on our Tech Support Hotline and help you quickly solve any technical problems you may encounter. Call Lifeboat for the complete solution to your programming needs.

LANGUAGES

Lattice C 3.0 The best selling C compiler has been upgraded to give you more functions and features. Lattice C 3.0 contains 200 new library functions, better code generation, support for new data types (void, enum, unsigned char, unsigned long), support for the 80186/80286 instruction set and the ability to generate in-line 8087/80287 instructions. Lattice C is the C compiler for professional developers.

RUN/C—The C Interpreter Learn C the natural way with RUN/C. The user interface is similar to BASIC with easy, familiar commands. The new 2.0 version of RUN/C comes with a full-screen editor and other enhancements.

RUN/C Professional New RUN/C Professional "...is the overall best choice of a C interpreter." *PC Tech Journal* (5/86). All RUN/C's capabilities plus powerful features for program development. Load your favorite object libraries with RUN/C Professional. Contains a full-screen editor and source code debugging facilities.

Pro Pascal A truly standard Pascal. Produces fast, tight code with plenty of compile time options.

BetterBASIC New Version Now you can program in BASIC and use the full memory of your PC, create structured programs using functions and procedures, make your own library modules and more. Now compatible with Microsoft BASIC.

LANGUAGE UTILITIES

Plink86 Plus An overlay linkage editor for linking 8086/8088 object modules. Contains new features for memory caching, library allocation, file merging and overlay reloading.

Pfix86 Plus A symbolic and source level advanced debugger for programming professionals.

Periscope I & II The symbolic debugger that can debug anywhere within a program at anytime with a simple press of a button. Periscope's breakout button will interrupt infinite loops, gain freedom from a locked up keyboard or help you find out where that slow running program is spending all its time. Periscope I additionally features a protected memory board which shields the debugger from runaway programs.

BASTOC A BASIC to C translator for the BASIC programmer who wants to upgrade to C.

EDITORS

Brief This programmer's editor has several outstanding features. Brief's macro language compiler allows the programmer to create new commands and assign them to any key. Brief's unique "undo" facility will even let you recover from accidentally entered commands. Create as many windows as will fit on a screen, containing the same or different portions of one or several files. Compile programs without ever having to leave Brief.

FirsTime Speed-up program development with an easy-to-use C and Pascal syntax checking editor. Detects syntax errors, undefined variables and mismatched type assignments. Use function key menus to generate statements or enter your code directly. On-line help available.

VEDIT Plus A full-screen text editor for program development and word processing. It contains powerful features including use of macros, on-line help facility, paragraph formatting, and file comparison.

EMACS Customizable editor including windowing, multi-tasking and special modes for C and Pascal.

FUNCTIONS

The Greenleaf Comm Library A library of over 120 communication routines. Contains functions to create interrupt driven routines or perform direct I/O to multiple Comm Ports. Its strengths are in asynchronous communications, interrupt mode, modem control, XMODEM, XON/XOFF and flow control.

PforCe New A library of 400 plus functions for interrupt driven communications, background tasks, string/table parsing, data base manipulation, DOS and BIOS functions and more. Superbly written and documented software with complete source code included.

The Greenleaf Functions A mature library of over 200 functions. Version 3.0 offers all new indexed documentation, with an abundance of examples. Source code included.

Essential C Utility Library Over 300 functions, with special attention given to screen handling, windows and business graphics. Source code is included.

BASIC_C Speed-up development time with this BASIC to C library, which acts as a bridge to C providing many of the capabilities of the BASIC language.

GRAPHICS, WINDOWING and SCREEN DESIGN

GSS*CGI GRAPHICS New The GSS Computer Graphics Interface is designed for creating high performance graphics-based applications. GSS*CGI speeds up application development and provides compatibility with a wide range of peripherals. It's the only CGI implementation that provides true device-independence for both raster and vector graphics. Products in the GSS GRAPHICS line include: the GSS*CGI Graphics Development Toolkit, the GSS Kernel System; the GSS Plotting System; and the GSS*CGI Metafile Interpreter.

Essential Graphics New A brand new graphics library for C programmers with the emphasis placed on ease-of-use and portability. No royalties.

WINDOWS FOR DATA New Provides C programmers with complete control over screen display and data entry, within an easy-to-use windowing system. Features include one-step data entry, multiple data types, Lotus-style menu design, full-featured field editing, pop-up entry windows, and much more.

Panel A powerful tool for interactive screen design.

For more information on these and other products in our complete line call:

1-800-847-7078

NY: 914-332-1875

Special Introductory Offer! For a limited time only you can get the new Software Programming Tools Guide from Lifeboat **FREE** by returning this coupon. To receive your copy of this regular $9.95 value, you must supply all of the information below. Sophisticated programmers won't want to be without this complete guide to product features and selection.

P

Name ____________ Title ____________
Company ____________ Business phone ____________
Address ____________ Zip ____________

The Full-Service Source for Programming Software

LIFEBOAT

INTERNATIONAL SALES OFFICES

Australia:
Fagan Microprocessor Associates
Phone: (61) 3699-9899
Canada: Scantel Systems
Phone: (416) 449-9252
England: Grey Matter, Ltd.
Phone: (44) 364-53499
Italy: Lifeboat Associates, S.p.A.
Phone: (02) 656-841
Japan: Lifeboat Japan
Phone: (03) 293-4711
Spain: Micronet, S.A.
Phone: (34) 1-457-5056
The Netherlands:
GIGA Computer Products
Phone: (31) 10-771846
West Germany:
MEMA Computer GmbH
Phone: (49) (069)34 72 26
Omnitex
Phone: (49) (076) 23 61820

Lifeboat
55 South Broadway
Tarrytown, NY 10591

CIRCLE NO. 144 ON READER SERVICE CARD

```
        lodsw                       ;get screen buffer offset
        mov     [si+erase_screen_offset-draw_screen_offset-2],ax
                                    ;save what will be old position
        mov     di,[si+draw_image_offset-draw_screen_offset-2]
                                    ;save what will be old image offset
        mov     [si+erase_image_offset-draw_screen_offset-2],di
        cmp     ax,0ffffh           ;see if object is active
        je      skip_this_object2 ; jmp if it isn't
        mov     di,ax               ;save screen buffer adr
        lodsw                       ;get distance to odd scan line
        mov     bp,ax               ; save in BP for inline code use
        lodsw                       ;get distance to even scan line
        mov     dx,ax               ; save in DX for inline code use
        add     si,length_of_erase_parms ;skip the erase parameters
        lodsw                       ;get the draw inline row adr
        mov     cx,ax               ; save in CX for inline use
        lodsw                       ;get the draw inline row adr
        mov     bx,ax               ; save in BX for call to inline
        lodsw                       ;get draw_image offset
        push    si                  ;save pointer to next queue item
        mov     si,ax               ;save draw image in SI
        call    bx                  ;draw it!
        pop     si                  ;restore where we left off in queue
        cmp     si,offset end_of_queue      ;see if we are done
        jne     get_next_objects_screen_adr2 ; jmp if more to erase
;
finish_up:
if ega
        cli
        mov     al,20h  ;issue a non_specific EOI (End Of Interrupt)
        out     20h,al  ; so that interrupt controller chip will
                        ; acknowledge future vertical interrupts
        mov     dx,cs:[crtc_base_address] ;re-enable interrupt
        mov     al,11h  ; select vertical retrace end
        out     dx,al   ;  register and clear vertical interrupt
        mov     al,04h  ;
        inc     dx      ;
        out     dx,al   ;  then enable vertical interrupt
        mov     al,14h  ;
        out     dx,al   ;
endif
        pop     es      ;restore all registers
        pop     ds      ;
        pop     di      ;
        pop     si      ;
        pop     bp      ;
        pop     cx      ;
        pop     bx      ;
        pop     ax      ;
        pop     dx      ;
        mov     cs:[need_to_draw_something_flag],false
                        ;indicate no reason to  draw again until the
                        ; queue is changed
        iret            ;restore flags and contine where interrupted
;
skip_this_object2:
        add     si,(queue_item_length-2)     ;point SI to next item
        cmp     si,offset end_of_queue       ;see if we are done
        jne     get_next_objects_screen_adr2 ; jmp if not
        jmp short finish_up                  ; jmp if all finished
put_objects_on_screen endp
;
;
;This is inline code for finding the screen address for each line
; of the image and calling the AND-OR inline code.
;
rlabel   macro   xx     ;this macro is used to label the inline code
rline&xx&:              ; entry points
        endm
;
;
; inline code for rows
;
xx=42                   ;there will be an entry point for each even
                        ; number of lines between 2 and 40. They will
                        ; be labeled "rline2", "rline4", ... "rline40"
        rept    20      ;each repeat handles two lines
xx=xx-2                 ;calculate number of lines for entry point
        rlabel  %xx     ;put in label for entry point
```

```
        call    cx      ;CX holds address of inline columns code
        add     di,bp   ;calculate the address to start next line
        call    cx      ;process image for odd scan line
        sub     di,dx   ;calculate the address to start next line
        endm            ; the next line will be an even line
        ret
;
;Inline code for AND-ORing a line of the image into the screen
;
clabel   macro  xx      ;this macro is used to label the inline code
cline&xx&:              ; entry points for number of columns to AND-OR
        endm            ;
;
xx=10                   ;this code can handle an image up to ten words
        rept    10      ; wide
        clabel  %xx     ;put in label for entry based on number of
                        ; words in a column
        lodsw              ;get mask word
        and     ax,es:[di] ;mask out background
        or      ax,[si]    ;insert data word
        inc     si         ;point to next mask word
        inc     si         ;
        stosw              ;return modified word to memory
xx=xx-1                    ;adjust label number
        endm
        ret                ;this return is executed at the end of every
                           ; line
;
;This table is used as an indirect address for jumping into
; the inline code for image moving.
;
row_inline_vector_table label word ;there is no entry point for zero
                                   ; lines. Starting at 2 eliminates
                                   ; the need to store a dummy entry
                                   ; point address
row_entry_address   macro   xx     ;this macro is used to generate
              dw      rline&xx&    ; the labels corresponding to the
              endm                 ; inline code entry points
;
xx=2
        rept    20
        row_entry_address       %xx
xx=xx+2
        endm
;
;This table is used as an indirect address for jumping into
; the inline code for exclusive-ORing columns.
;
column_inline_vector_table label word ;there is no entry point for zero
                                      ; lines. Starting at 2 eliminates
                                      ; the need to store a dummy entry
                                      ; point address
column_entry_address   macro   xx     ;this macro is used to generate
               dw      cline&xx&      ; the labels corresponding to the
               endm                   ; inline code entry points
;
xx=1
        rept    10
        column_entry_address   %xx
xx=xx+1
        endm
;
;This is inline code for erasing the image by restoring the screen
; memory map from the background buffer.
;
elabel  macro   xx      ;this macro is used to label the inline code
eline&xx&:              ; entry points
        endm
;
xx=42              ;there will be an entry point for each even
                   ; number of lines between 2 and 40. They will
                   ; be labeled "eline2", "eline4", ... "eline40"
    rept    20
xx=xx-2            ;calculate number of lines for this entry point
    elabel  %xx    ;put in label for entry point
    mov     si,di  ;erase using same offset in background buffer
    mov     cx,ax  ;put width of image in words in CX to prepare for
    rep     movsw  ; repeated move string on even line
    add     di,bp  ;calculate address of next line DI + (2000h-width)
```

```
     mov      si,di    ;erase using same offset in background buffer
     mov      cx,ax    ;put width of image in bytes in CX to prepare for
     rep      movsw    ; repeated move string on odd line
     sub      di,dx    ;calculate address of next line DI - (1fb0h+width)
     endm
     ret
;
;This table is used as an indirect address for jumping into
; the inline code for erasing an image.
erase_inline_vector_table label word ;there is no entry point for zero
                                      ; lines. Starting at 2 eliminates
                                      ; the need to store a dummy entry
                                      ; point address
entry_address    macro    xx          ;this macro is used to generate
                 dw       eline&xx&   ; the labels corresponding to the
                 endm                 ; inline code entry points
;
xx=2
        rept    20
        entry_address   %xx
xx=xx+2
        endm
cseg    ends
        end
```

LISTING 3: SPRITES3.ASM

```
;Byte-move graphics driver for putting rectangular images into
; the Color/Graphics Adapter's medium-resolution memory map.
;
; Inputs (DS and CS must be the same, ES must contain 0B800h):
;   BX - Row at which to put the top of the image. Range is 0 to 198.
;        BX must be even: images cannot start at an odd row.
;   CX - The column at which to start the image in bytes. Range is
;        0 to 79.
;   SI - Points to data structure which contains the image description.
;        The first byte is the height in lines (h). The second is the
;        width of the image in bytes (w). h*w image bytes follow which
;        contain the values to be placed into the screen memory map.
;        The image bytes are ordered as h consecutive sets of w line
;        image bytes.  h must be an even number; images must be an
;        even number of scan lines high.
; Outputs: None. AX,BX,CX,DX,BP,SI,DI destroyed.
;
cseg    segment para public 'cseg'
        assume  cs:cseg,ds:cseg,es:nothing
        public  byte_move_form_driver
;
byte_move_form_driver proc near
        mov     di,[bx+even_line_screen_offset_table]  ;find the offset
                                                         ; of the top
                                                         ; line of image
        add     di,cx     ;ES:DI now points to byte at which to put
                          ; the image's upper left corner
        lodsb             ;get the height of the image
        xor     ah,ah     ;make height a word value for calculations
        mov     bx,ax     ;store height in BX
        lodsb             ;get the width of the image in bytes
        mov     bp,2000h  ;calculate the amount to add after even scan
        sub     bp,ax     ; lines are drawn to get to the address of the
                          ; next scan line
        mov     dx,1fb0h  ;calculate the amount to subtract after odd
        add     dx,ax     ; scan lines are drawn to get to the address
                          ; of the next scan line
        jmp     [bx+inline_height_vector_table-2]  ;-2 because there is
                                                    ; no zero lines entry
                                                    ; point
;This table is used to find the offset of an even scan line in
; the memory map of the color graphics adapter in medium
; resolution mode.
;
even_line_screen_offset_table label word
x=0
        rept    100      ;there are 100 even lines
        dw      x*50h    ; each is 50h (80 decimal) long
x=x+1
        endm
;
;This is inline code for moving the image into the screen memory map.
;
label   macro   x       ;this macro is used to label the inline code
line&x&:                ; entry points
        endm
;
x=42                    ;there will be an entry point for each even
                        ; number of lines between 2 and 40. They will
                        ; be labeled "line2", "line4", ... "line40"
        rept    20
x=x-2                   ;calculate number of lines for this entry point
        label   %x      ;put in label for entry point
        mov     cx,ax   ;put width of image in bytes in CX to prepare
        rep     movsb   ; for repeated move string on even line
        add     di,bp   ;calc address of next line DI + (2000h-width)
        mov     cx,ax   ;put width of image in bytes in CX to prepare
        rep     movsb   ; for repeated move string on odd line
        sub     di,dx   ;calc address of next line DI - (1fb0h+width)
        endm
        ret
;
;This table is used as an indirect address for jumping into
; the inline code for image moving.
;
inline_height_vector_table label word ;there is no entry point for zero
                                       ; lines. Starting at 2 eliminates
                                       ; the need to store a dummy entry
                                       ; point address
entry_address    macro    x            ;this macro is used to generate
                 dw       line&x&      ; the labels corresponding to
                 endm                  ; the inline code entry points
;
x=2
        rept    20
        entry_address   %x
x=x+2
        endm
byte_move_form_driver endp
cseg    ends
        end
```

Save 50%

PC TECH JOURNAL

Subscribe to the only IBM PC magazine written for systems designers and integrators **and save 50%**

Enter my subscription to PC TECH JOURNAL for:

☐ One year (13 issues) for only $26.70.

☐ Two years for only $53.35. **SAVE 50%!**

Savings based on annual single copy price of $53.35.

Mr./Mrs./Ms. ______________________________
please print name in full 4S507

Company ______________________________

Address ______________________________

City ______________________________

State ______________ Zip ______________

Add $6. per year for postage in Canada and all other foreign countries—Cash payment U.S. currency only. Basic annual subscription price is $34.97.
Please allow up to 30 to 60 days for delivery of first issue. *Send no money—we will bill you later!*

Save 50%

PC TECH JOURNAL

Subscribe to the only IBM PC magazine written for systems designers and integrators **and save 50%**

Enter my subscription to PC TECH JOURNAL for:

☐ One year (13 issues) for only $26.70.

☐ Two years for only $53.35. **SAVE 50%!**

Savings based on annual single copy price of $53.35.

Mr./Mrs./Ms. ______________________________
please print name in full 4S507

Company ______________________________

Address ______________________________

City ______________________________

State ______________ Zip ______________

Add $6. per year for postage in Canada and all other foreign countries—Cash payment U.S. currency only. Basic annual subscription price is $34.97.
Please allow up to 30 to 60 days for delivery of first issue. *Send no money—we will bill you later!*

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT #66 BOULDER, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

TECH JOURNAL
P.O. BOX 2966
BOULDER, COLORADO 80321

Sixteen-color Graphics

A little-known low-resolution mode of the IBM CGA makes graphics possible in a full 16 colors.

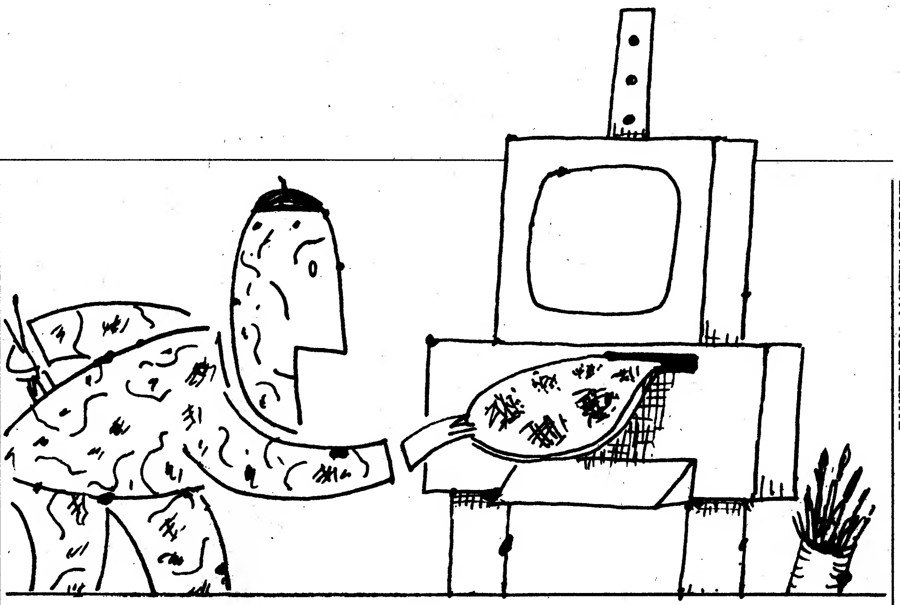


ILLUSTRATION • MACIEK ALBRECHT

The Motorola 6845 CRT controller chip on the IBM Color Graphics Adapter (CGA) is, according to the IBM *Technical Reference Manual*, "highly programmable with respect to raster and character parameters. Therefore, many additional modes are possible with clever programming...." One of these modes can be used to add to the four graphics colors of the CGA, providing 16-color graphics—albeit in low resolution (80 by 100).

The resolution of this mode can be increased to 160 by 100 using a software technique that involves splitting the character block into two smaller pixel blocks. Although the character cell in this text mode is 2 pixels high and 8 pixels wide, making the definition of readable characters on the screen quite difficult, it nonetheless can be used to create striking graphics (see photo 1). The Enhanced Graphics Adapter (EGA) does not provide support this mode; however, this point is academic, because a 16-color graphics mode already is provided in the EGA.

The techniques for splitting one character block into two pixel blocks have been implemented in a series of Turbo Pascal procedures that provide the necessary tools for producing low-resolution graphics. These procedures are combined in the sample program MANDEL.PAS to allow the user to draw the Mandelbrot set, a technique featured in the August 1985 issue of *Scientific American* ("Computer Recreations," A. K. Dewdney, p. 16).

Associated with each character cell on the text screen in the low-resoltuion mode is an attribute byte containing the foreground and background color information. Four bits are allocated to each color; thus, 16 possibilities are available for both foreground and background. (The high bit of the background nibble typically controls blinking, leaving three bits to control color. As a result, only eight colors are possible. However, a mode register setting is available that instructs the CGA to use the fourth bit to control intensity. This yields 16 colors for background as well.)

Because foreground and background colors are independent of each other, the 2-by-8 character cell can be divided into two 2-by-4 cells if the *half-block* character 222 (▐) is used to place a point on the screen. To plot the point (100,30) in light red, the text cursor is placed at (50,30) and character 222 is written with background 12 and foreground 0. The point at (101,30) can be plotted in dark blue by writing character 222 at cursor location (50,30) with foreground 1 and background 0.

IBM does not document support for this mode in the IBM *Technical Reference Manual*. Nonetheless, the BIOS can be used in this mode to position a cursor, to read the current attribute of a point, and to plot a point in a given color. Using the BIOS in a CGA driver prevents screen interference (snow) and results in shorter programs. Unfortunately, BIOS calls proved too slow for this application; thus, the program shown in listing 1 (MANDEL.PAS) writes directly to screen memory. Because of this low-level access, both MANDEL.PAS and the alternate version of the program, which is called MANDEL87.PAS (listing 2), will hang an EGA.

A complication occurs when plotting a point. While the foreground and background colors of a cell are independent, they both are contained in the same attribute byte. The user must be sure, when modifying the foreground to plot one pixel, not to change the background and affect the adjacent pixel as well. To plot a new foreground color, the user masks off the old foreground color from the appropriate attribute byte and adds the new foreground color. Changing the background color is more complicated, because the new color must be shifted to the left four bits. (See figure 1 for examples.)

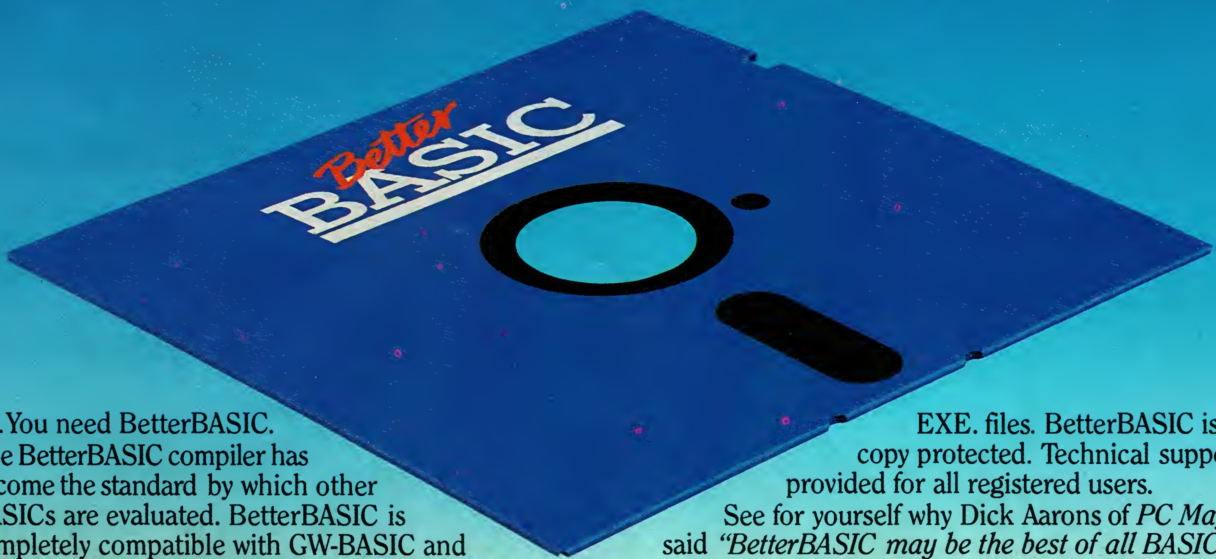
The mathematical ideas upon which MANDEL.PAS is based are relatively simple. In the plane of complex numbers, the function $f(z) = z^2$ is used to describe a simple iterative process. For any given point p the sequence z_0, z_1, z_2,... can be generated using the following three equations:

$z_0 = p$
$z_1 = f(z_0) + p$
$z_2 = f(z_1) + p$,...

Beginning with $p = 1+2i$, for example, the result is:

$z_0 = 1+2i$
$z_1 = (1+2i)^2 + (1+2i) = -2+6i$
$z_2 = (-2+6i)^2 + (1+2i) = (-31-22i)$,...

WHEN YOU NEED ACCESS TO FULL MEMORY, STRUCTURE, COMPATIBILITY WITH GW- & PC-BASICA, OR WHEN YOU NEED THE POWER AND FLEXIBILITY OF C OR PASCAL...



. . .You need BetterBASIC. The BetterBASIC compiler has become the standard by which other BASICs are evaluated. BetterBASIC is completely compatible with GW-BASIC and PC-BASICA when running on IBM PC's and true clones. You can load and run your existing BASIC programs in BetterBASIC. It uses standard MicroSoft syntax and gives you more than 150 additional statements such as XREF, DEFINE WINDOW, MAKE MODULE, and PROCEDURE. In benchmark comparisons, BetterBASIC is five times faster than interpreted BASIC. There is optional 8087/80287 math chip support, and an optional Runtime System to create stand-alone EXE. files. BetterBASIC is not copy protected. Technical support is provided for all registered users.

See for yourself why Dick Aarons of *PC Magazine* said *"BetterBASIC may be the best of all BASIC programming worlds"* and selected BetterBASIC as "Editor's Choice" (Oct. 29, 1985).

BetterBASIC	$199
8087/80287 Math Chip Support	$99
Runtime System	$250
Sample Disk with Tutorial	$10

Ask your dealer or call to order: **1-800-225-5800**

In Canada, call 416-469-5244

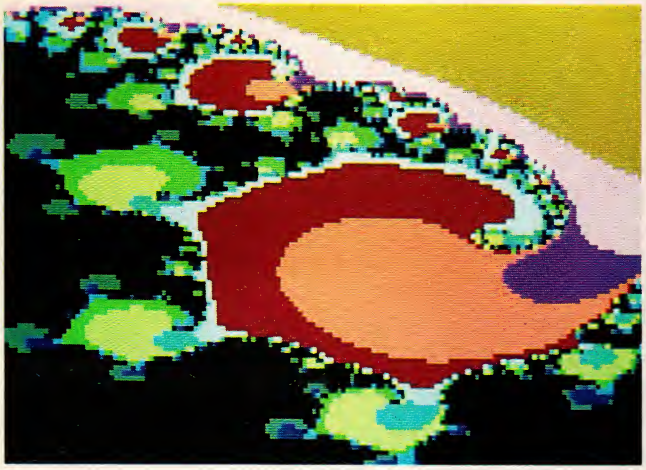
Better BASIC™

BetterBASIC is a registered trademark of Summit Software Technology, Inc. IBM PC, XT, AT are registered trademarks of International Business Machines Corp. Microsoft is a registered trademark of Microsoft Corp. Tandy is a registered trademark of Tandy Corp.

CIRCLE NO. 195 ON READER SERVICE CARD

Summit Software Technology, Inc.™
106 Access Rd. Norwood, MA 02062

PHOTO 1: *Mandelbrot Set Output*



A boundary area of the Mandelbrot Set is displayed in the hybrid 160-by-100 16-color mode. Each pixel represents a point on the complex plane; each pixel's color shows the point's stability under a mathematical transformation.

FIGURE 1: *Splitting Character Blocks*

	bbbb ffff	Old attribute
and	1111 0000	Mask
	bbbb 0000	Yield after and with mask
+	0000 nnnn	New color
	bbbb nnnn	New attribute
	bbbb ffff	Old attribute
and	0000 1111	Mask
	0000 ffff	Yield after and with mask
+	nnnn 0000	New color shifted to the left 4 bits
	nnnn ffff	New attribute

Horizontal resolution is doubled by treating each 8-bit character block as two 4-bit blocks. The four bits denoted *bbbb* control the background color (top), while the four bits labeled *ffff* control the foreground color (bottom).

The sequence generated at point *p* can wander to infinity, its real and/or imaginary parts becoming larger and larger as it proceeds. In this example the terms of the sequence do not become infinite. Because a computer cannot generate the entire sequence for each point, the distinction between a wandering and a nonwandering point is determined as follows: if the terms of a sequence do not exceed 2 in magnitude after a certain number of those terms has been computed (in this program, 64), the point *p* is in the nonwandering set. This distinction is not as arbitrary as it seems—there is a mathematical justification. (The magnitude of the complex number $a+bi$ is the square root of a^2+b^2. The time-consuming process of taking the square root can be avoided by determining instead if a^2+b^2 is greater than 4.)

If the point *p* is in the nonwandering set, it is plotted in black, and the

program continues to the next point. If the terms in the sequence at p eventually exceed 2 in magnitude, then the point is plotted in a color dependent on the number of terms computed before the magnitude exceeded 2. In this case, the advantage of a possible 16 colors from which to choose is obvious.

The program MANDEL.PAS contains four constants (see the CONST sections of listings 1 and 2), which are to be set by the user. The constants *xmin* and *ymin* determine the lower left corner of the domain to be considered; *xrange* and *yrange* determine the length and height of this domain. The entire region of interest for the function $f(z) = z^2$ is contained within the square defined by $(xmin, ymin) = (-2, -1.3)$ and $(xrange, yrange) = (2.6, 2.6)$. (Outside this square the sequence immediately exceeds 2 in magnitude.)

An interesting feature of the program MANDEL.PAS is its ability to magnify any selected portion of the domain. This enables the user to view the exceedingly intricate intertwining that occurs between the wandering and the nonwandering sets for this function. Table 1 lists the coordinates of areas of interest along the border of the graph of the Mandelbrot set.

TABLE 1: *Coordinate Sets*

XMIN	XRANGE	YMIN	YRANGE
−1.6	0.4	0	0.22
−0.29110	0.00004	0.85828	0.00003
−0.2910783	0.0000008	0.8582886	0.0000008
−0.291077855	0.000000015	0.858289110	0.000000015
0.2908	0.0050	−0.0200	0.0075
0.2941	0.0010	−0.0167	0.0010
−0.5575	0.0052	0.6220	0.0052

The four variables *xmin*, *xrange*, *ymin*, and *yrange* specify the location and size of windows into the graph of the Mandelbrot set. Especially interesting patterns are located at the coordinate sets listed above. The three sets of coordinates shown in the middle of the table demonstrate the zoom feature of the program.

The disadvantage of MANDEL.PAS is its execution time. Each of the 16,000 points requires 10 floating-point calculations to produce each element in its sequence and to compute the square of its magnitude. A point can require that the maximum number of sequence elements (64) be computed; thus, each point could require 640 floating-point operations. If each point requires 320 operations, the entire screen needs 5 million floating-point computations to complete. The 8087 coprocessor would be advantageous.

MANDEL87.PAS is a version of MANDEL.PAS in which Iterate(x,y) has written INLINE using 8087 code. This program requires one-twentieth of the time that is required by MANDEL.PAS to complete a typical screen. In addition, it reduces screen-draw time to two minutes on a 6-MHz PC/AT.

Richard Chandler and Gary Faulkner hold doctorates in mathematics and currently are teaching at North Carolina State University. Michael Davis is a graduate student in mathematics at that university.

LISTING 1: MANDEL.PAS

```
Program Mandelbrot;

type
   reg        = array[0..11] of byte;
const
   xmin       = -2.0;
   xrange     =  2.6;
   ymin       = -1.3;
   yrange     =  2.6;
   crt_index_reg   =$03D4; { Port # of Index register of 6845        }
   crt_data_reg    =$03D5; { Port # of Input register of 6845        }
   mode_select_reg =$03D8; { Port # of video mode select register    }
   color_select_reg=$03D9; { Port # of video color select register   }
var
   c,j,k,n       : integer;
   x,y,dx,dy     : real;
   crt_mode_set
         : byte absolute $0000:$0465; { Used by BIOS to maintain    }
   crt_palette
         : byte absolute $0000:$0466; { values of mode & color regs }
   screen        : array[1..16384] of byte absolute $B800:$0000;
label
   quit;

{------- CLEARS SCREEN -------------------------------------------}

Procedure ClearScreen;
begin
   port[mode_select_reg] := 0; { Disables video to prevent snow      }
   FillChar(screen,16384,0);   { Fills screen with chr 0 attribute 0}
   port[mode_select_reg] := 9; { Enables video to see screen         }
end;

{------- SET 6845 CRT CONTROLLER TO LO-RES MODE ------------------}

Procedure LoRes;
const
   regdata  :  reg = (113,80,90,10,127,6,100,112,2,1,32,0);
var
   i        :  byte;
begin
   crt_mode_set := 0;
   crt_palette  := 0;
   port[color_select_reg] := 0;
   for i := 0 to 11 do
   begin
      port[crt_index_reg] := i;
      port[crt_data_reg]  := regdata[i];
   end;
   ClearScreen;
   crt_mode_set := 9;
end;
{------- SET 6845 CRT CONTROLLER TO 80x25 TEXT SCREEN ------------}

Procedure TextScreen;
const
    regdata  :  reg = (113,80,90,10,31,6,25,28,2,7,6,7);
var
   i         :  byte;
begin
   for i := 0 to 11 do
   begin
      port[crt_index_reg] := i;
      port[crt_data_reg]  := regdata[i];
   end;
   crt_mode_set := 41;
   ClrScr;
end;

{------- PLOTS POINT AT (x,y) in COLOR c -------------------------}

Procedure Point(x,y,c:integer);
begin
   inline($B8/$00/$02/     { MOV AX,0200H   0200 -> AX       }
```

Powerful MS™-DOS Software.

For the IBM®-PC, XT, AT & others with generic MS-DOS/PC-DOS 2.0 or higher.

Not copy protected!

Only $39.95 Each!

UTAH COBOL™

Whether student, teacher or professional programmer, this is the one you've heard so much about.

- ☐ It's easy to use. Compiles 5000 statements on a 128K machine.
- ☐ 170 clear error messages, i.e. DATA-NAME IS MISSING OR MISSPELLED.
- ☐ Distribute your object code programs royalty free.
- ☐ Small object code programs conserve disk space.
- ☐ Fast compile times to increase programmer productivity. Over 25 times faster than one compiler costing $995!
- ☐ You get a diskette and 213-page manual with lots of examples and 16 complete COBOL source code programs. **$39.95.**

Also available: COBOL Application Packages, Book 1 **$9.95.**

UTAH PASCAL™

- ☐ 14-digit precision, BCD math, no round-off errors with decimal arithmetic for business and floating point +63 −64 for scientific.
- ☐ A very nice TRACE style debugging.
- ☐ Arrays up to 8 dimensions and 64K strings.
- ☐ External procedures and functions with dynamic auto-loading.
- ☐ One-step compile, no assembly or link required.
- ☐ You get a 132-page manual and diskette. **$39.95**

UTAH PILOT™

- ☐ Perfect for industrial training, office training, drill and testing, virtually all programmed instruction, word puzzle games, and data entry facilitated by prompts.
- ☐ John Starkweather, Ph.D., the inventor of the PILOT language, has added a built-in full-screen text editor, and much more.
- ☐ Meets all PILOT-73 standards for full compatibility with older versions.
- ☐ You get a diskette, 125-page manual and ten useful sample programs. **$39.95.**

Also still available for 8-bit machines with CP/M® is our world famous Nevada Software Series used by 50,000 customers in 40 countries. These include Nevada COBOL, Nevada FORTRAN, Nevada PASCAL, Nevada PILOT, Nevada BASIC and Nevada EDIT. **$39.95** each.

Satisfaction guaranteed. If for any reason you're not completely satisfied, just return the package within 15 days in good condition, and we'll refund your money.

UTAH FORTRAN™

- ☐ FORTRAN IV based upon ANSI-66 standards.
- ☐ Very fast compile times and easy to use.
- ☐ IF.. THEN.. ELSE constructs.
- ☐ Chaining with blank and named common.
- ☐ Copy statement.
- ☐ ENCODE and DECODE.
- ☐ Free-format input and output.
- ☐ A very nice TRACE style debugging.
- ☐ 150 English language error messages.
- ☐ You get a diskette, and 223-page manual. **$39.95**

UTAH EDIT™

- ☐ A character-oriented full-screen video display text editor designed specifically to create COBOL, FORTRAN and PASCAL programs.
- ☐ Only requires 15K disk space so it can fit on the same disk as your compilers.
- ☐ Completely customizable tab stops, default file type, keyboard control key layout and CRT by menu selection.
- ☐ Diskette comes with easy to read 58-page manual. **$39.95.**

UTAH BASIC™

- ☐ This interpreter has a built-in full-screen editor.
- ☐ Single- and Multi-line user definable functions.
- ☐ BCD Math- no round-off errors.
- ☐ Full Matrix operations.
- ☐ You get 220-page manual and diskette. **$39.95.**

Handling/Shipping: No shipping charge within US. Overseas add $10 for first package, $5 each additional. Checks must be in US Dollars, drawn on a US bank.

Utah Software requires 128K RAM and PC-DOS or MS-DOS 2.0 or higher.

HOW TO ORDER. Send check or money order to Ellis Computing, Inc. with VISA or MASTERCARD order by phone. Sorry no COD's.

Ellis Computing, Inc.
5655 Riggins Court, Suite 10
Reno, Nevada 89502
Phone (702) 827-3030

SINCE 1977

ELLIS COMPUTING™

CIRCLE NO. 170 ON READER SERVICE CARD

```
            $30/$FF/          { XOR BH,BH       0 -> BH            }
            $8A/$56/$08/      { MOV DL,[BP+8]   x -> DL            }
            $D0/$EA/          { SHR DL,1        x/2->DL,rem->CF    }
            $8A/$76/$06/      { MOV DH,[BP+6]   y -> DH            }
            $CD/$10/          { INT 10H         locates cursor     }
            $B8/$00/$08/      { MOV AX,0800H    0800 -> AX         }
            $CD/$10/          { INT 10H         read attribute     }
            $8A/$5E/$04/      { MOV BL,[BP+4]   c -> BL            }
            $73/$05/          { JNC +5          x even => CF=0     }
            $25/$00/$F0/      { AND AH,F0H      discard old fg     }
            $EB/$0B/          { JMP +11         Jmp to col asmb    }
            $D0/$E3/          { SHL BL,1        x even so          }
            $D0/$E3/          { SHL BL,1         c is bg           }
            $D0/$E3/          { SHL BL,1          shift bg         }
            $D0/$E3/          { SHL BL,1           left 4 bits     }
            $25/$00/$0F/      { AND AH,0FH      discard old bg     }
            $00/$E3/          { ADD BL,AH       assemble color     }
            $B8/$DE/$09/      { MOV AX,09DE     chr ß to AH        }
            $B9/$01/$00/      { MOV CX,01       one to write       }
            $CD/$10);         { INT 10H         write chr, attr    }
end;

{------- DETERMINE NUMBER OF ITERATIONS AT (x,y) ----------------}

Function Iterate(x,y:real):integer;
var
   n          : integer;
   i,r,zi,zr  : real;
begin
   zr := x; zi := y;                          { Initialize z            }
   n := 64;                                   { Iteration counter       }
   repeat
      n := n-1;                               { Decrement counter       }
      r := zr*zr - zi*zi + x;                 { Real part of next z     }
      i := 2*zr*zi + y;                       { Imaginary part of next z}
      zr := r; zi := i;                       { Update z                }
   until (zr*zr + zi*zi > 4) or (n = 0);
                                      { Modulus» > 4 or 64 iterations   }
   Iterate := n;                      { Return 64 - # of iterations    }
end;

{------- MAIN PROGRAM BEGINS ------------------------------------}

begin
   LoRes;                                   { Switch to LoRes mode          }
   dx := xrange/159; dy := yrange/99;       { Scale world to screen         }
   y := ymin + yrange;                      { Maximum y to top of screen    }
   for j := 0 to 99 do                      { 100 rows on LoRes screen      }
   begin
     x := xmin;                             { Minimum x to left of screen   }
     for k := 0 to 159 do                   { 160 columns on LoRes screen   }
      begin
       n := Iterate(x,y);                   { Determine number of iterations }
       c := n div 8;                        { Determine color number 0..7   }
       if n mod 8 > 3 then c := c+8;        { If remainder = 4..7 then bright}
       Point(k,j,c);                        { Plot point on screen          }
       if keypressed then goto quit;        { Press any key to interrupt/quit}
       x := x + dx;                         { Update x coordinate of point  }
      end;                                  { Loop until finished with row  }
     y := y - dy;                           { Update y coordinate of point  }
   end;                                     { Loop until finished with screen}
quit:   repeat until keypressed;            { Hold picture until key pressed }
   TextScreen;                              { Restore normal text screen    }
end.
```

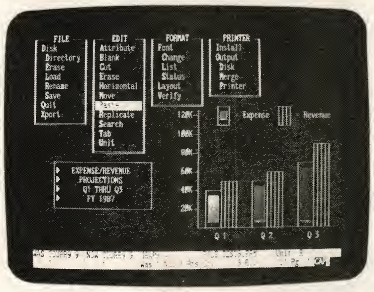
LISTING 2: MANDEL87.PAS

```
Program Mandelbrot87;

type
   reg       = array[0..11] of byte;
const
   xmin      = -2.0;
   xrange    =  2.6;
   ymin      = -1.3;
```

Now it costs even less to look good.

The Cordata Desktop Printshop™ has always been the best value in PC-compatible laser printers.

With speed and flexibility that the others can't match. Type fonts on diskette, not expensive cartridges. Data transfer up to 30 times faster than Hewlett-Packard's LaserJet+. Features implemented in software, for easy update. And Epson™ emulation, so it runs with most MS-DOS programs.



Not to mention being easier.

But now the Desktop Printshop is an even better value:

A menu-driven forms design software program for rapid document creation, the industry-acclaimed Wordstar 2000™ word processing program, and 38 fonts on diskette are included with the Desktop Printshop, for the amazingly low price of $2995.

So don't settle for just any laser printer. Demand the Desktop Printshop.

All it takes is a call to Cordata at: (800) 621-6746 (in CA: (805) 495-5800; in Canada: (604) 984-0641) for the name of your nearest Desktop Printshop dealer.

After all, how can you afford *not* to look your best?

The Desktop Printshop.™

CORDATA, 275 E. Hillcrest Drive, Thousand Oaks, CA 91360
TELEX 650-2696270; in Europe: Holland 032-18111

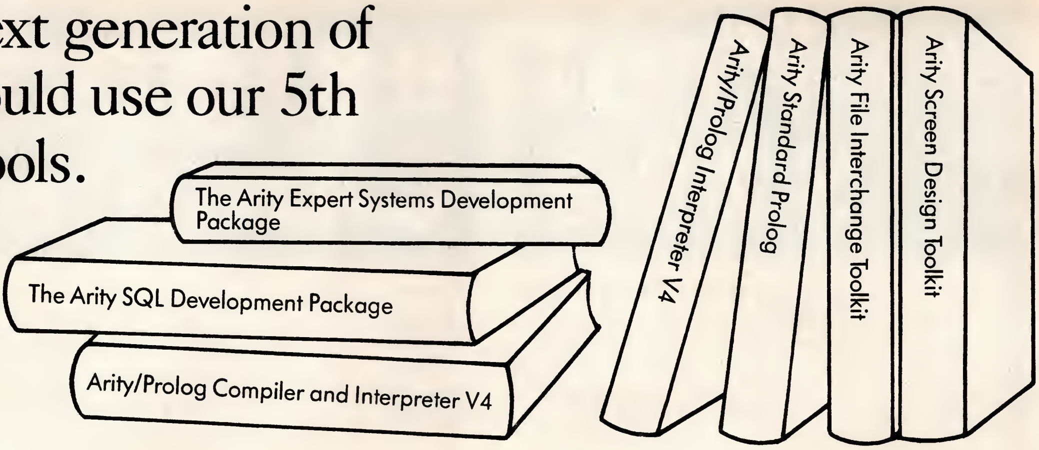
cordata

formerly corona data systems, inc.

Wordstar 2000 is a trademark of MicroPro International Corp. Epson is a trademark of Epson America, Inc.

```
  yrange     =  2.6;
 crt_index_reg     = $03D4; { Port # of Index register of 6845      }
 crt_data_reg      = $03D5; { Port # of Input register of 6845      }
 mode_select_reg   = $03D8; { Port # of video mode select register  }
 color_select_reg  = $03D9; { Port # of video color select register }
var
  c,j,k,n      : integer;
  x,y,dx,dy    : real;
  crt_mode_set : byte absolute $0000:$0465;
  crt_palette  : byte absolute $0000:$0466;
          { Used by BIOS to maintain values of mode & color regs }
  screen       : array[1..16384] of byte absolute $B800:$0000;
label
  quit;

{------- CLEARS SCREEN ------------------------------------------}

Procedure ClearScreen;
begin
  port[mode_select_reg] := 0; { Disables video to prevent snow     }
  FillChar(screen,16384,0);   { Fills screen with chr 0 attribute 0 }
  port[mode_select_reg] := 9; { Enables video to see screen        }
end;

{------- SET 6845 CRT CONTROLLER TO LO-RES MODE ------------------}

Procedure LoRes;
const
  regdata  :  reg = (113,80,90,10,127,6,100,112,2,1,32,0);
var
  i        :  byte;
begin
  crt_mode_set := 0;
  crt_palette  := 0;
  port[color_select_reg] := 0;
  for i := 0 to 11 do
  begin
    port[crt_index_reg] := i;
    port[crt_data_reg]  := regdata[i];
  end;
  ClearScreen;
  crt_mode_set := 9;
end;

{------- SET 6845 CRT CONTROLLER TO 80x25 TEXT SCREEN ------------}
Procedure TextScreen;
const
  regdata  :  reg = (113,80,90,10,31,6,25,28,2,7,6,7);
var
  i        :  byte;
begin
  for i := 0 to 11 do
  begin
    port[crt_index_reg] := i;
    port[crt_data_reg]  := regdata[i];
  end;
  crt_mode_set := 41;
  ClrScr;
end;

{------- PLOTS POINT AT (x,y) in COLOR c -------------------------}

Procedure Point(x,y,c:integer);
begin
  inline($B8/$00/$02/       { MOV AX,0200H    0200 -> AX        }
         $30/$FF/           { XOR BH,BH       0 -> BH           }
         $8A/$56/$08/       { MOV DL,[BP+8]   x -> DL           }
         $D0/$EA/           { SHR DL,1        x/2->DL,rem->CF   }
         $8A/$76/$06/       { MOV DH,[BP+6]   y -> DH           }
         $CD/$10/           { INT 10H         locates cursor    }
         $B8/$00/$08/       { MOV AX,0800H    0800 -> AX        }
         $CD/$10/           { INT 10H         read attribute    }
         $8A/$5E/$04/       { MOV BL,[BP+4]   c -> BL           }
         $73/$05/           { JNC +5          x even => CF=0    }
         $25/$00/$F0/       { AND AH,F0H      discard old fg    }
         $EB/$0B/           { JMP +11         Jmp to col asmb   }
         $D0/$E3/           { SHL BL,1        x even so         }
         $D0/$E3/           { SHL BL,1         c is bg          }
         $D0/$E3/           { SHL BL,1          shift bg        }
```

Why your next generation of products should use our 5th generation tools.



Arity's integrated family of programming tools allows you to combine software written in Arity/Prolog, the best of the fifth generation languages, with Arity SQL, the best of the fourth generation languages, and with conventional third generation languages such as C or assembly language to build your smarter application.

You can use Arity/Prolog to build expert systems using the Arity Expert Systems Development Package. Or to build natural language frontends. Or to build intelligent information management systems. Arity/Prolog lets you build advanced technology into your vertical applications package.

And more...

That's not the whole story. Arity's products are all designed to be fast, powerful, serious. Each of our products contains unexpected bonuses. Such as a one gigabyte virtual database integrated into Arity/Prolog. The most powerful of its kind on a PC.

Quality first. Then price.

In order to be the best, we had to prove it to our customers. Our tradition of quality software design is reflected in every product we sell. Quality first. Then price. And we always provide the best in customer support.

Our products are not copy protected. We do not charge royalties. We offer generous educational and quantity discounts. And we have a 30 day money back guarantee.

Try us to know that we keep our promise on commitment to quality and reliability. Try us by using our electronic bulletin board at 617-369-5622 or call us by telephone—you can reach us at 617-371-2422.

Or fill in this coupon. Whether you order today or not, let us send you full descriptions of our integrated family of Arity products.

CIRCLE NO. 136 ON READER SERVICE CARD

arity

We design and distribute high quality, serious application software for the IBM PC, XT, AT and all MS-DOS compatibles.

Please complete this form to place your order and/or request detailed information.

		Quantity	Info only
Arity/Prolog Compiler and Interpreter V4	$795.00		
Arity/Prolog Interpreter V4	$350.00		
Arity Standard Prolog	$ 95.00		
Arity SQL Development Package	$295.00		
Arity Expert System Development Package	$295.00		
Arity Screen Design Toolkit	$ 49.95		
Arity File Interchange Toolkit	$ 49.95		
TOTAL AMOUNT (MA residents add 5% sales tax) (These prices include shipping to all U.S. cities)		$	

NAME__________

SHIPPING ADDRESS__________

CITY/STATE/ZIP__________

TELEPHONE__________

Payment: ☐ Check ☐ PO ☐ AMEX ☐ VISA ☐ MC

Card #__________ Exp. date__________

Signature__________

ARITY CORPORATION • 358 BAKER AVENUE • CONCORD, MA 01742

arity

ProDesign II
$299. Complete

Many CAD companies are now charging extra for features we consider necessary for a functional CAD system. For example, the price of the Basic AutoCAD System was recently lowered to $300. However, it costs $1700 to $2200 more to get such features as Auto Dimensioning, Area Fill, Mirroring, Filleting, and Curve Fitting. With ProDesign II, you get the entire CAD package for one low price — $299. Complete.

But there is another, very important reason you should get ProDesign II. ProDesign II is very unique among CAD packages. It is easy to learn and use. When we call ProDesign II "The Easy To Use CAD System", we're not joking. You will be productive with ProDesign II in an hour or two instead of a week or two.

What else does ProDesign II offer? Compatibility. Compatibility with over 100 printers. Compatibility with over 50 plotters. Compatibility with most graphics adapters, digitizing, tablets, and mice. Compatibility with most display adapters. Compatibility with any software that can produce HP plotter commands. Compatibility with mainframe CAD systems (IGES). All at no extra charge!

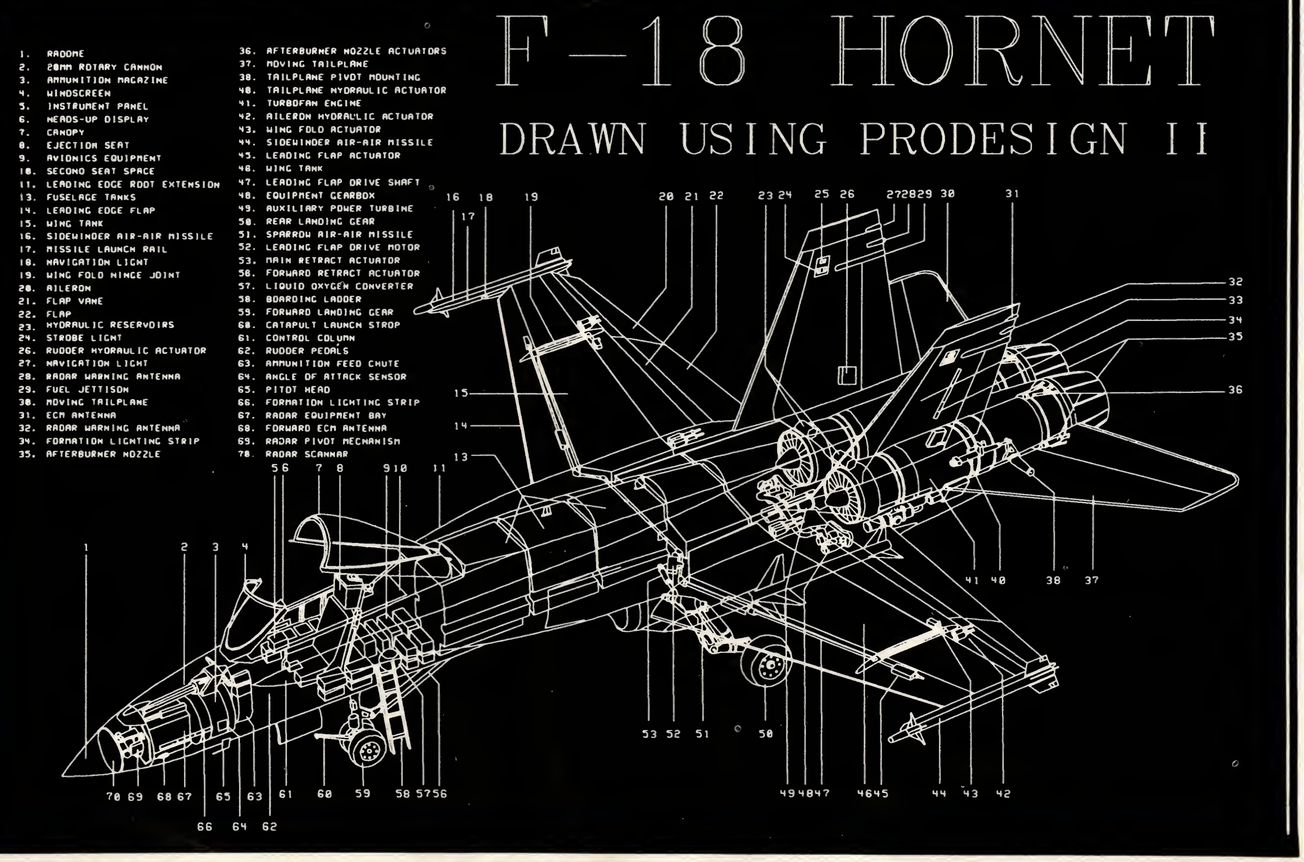
What additional hardware do you need? None! An IBM PC compatible and 512K RAM are all that you need. ProDesign II produces high resolution drawings on your dot matrix printer - better than .005" resolution on an IBM/Epson compatible printer. And you don't have to get a mouse or digitizing tablet. ProDesign II is designed for easy and efficient keyboard use. You won't even need a math coprocessor, although it is supported. From a small drawing on a dot matrix printer to an E-Size drawing on a large plotter, ProDesign II is the one to use!

ProDesign II — Affordable, Compatible, and Usable!

Where do you get it? See your local dealer or contact:
American Small Business Computers, Inc.
118 South Mill Street • Pryor, OK 74361
918/825-4844

	ProDesign II	AutoCAD
Auto Dimensioning	$299	$2000
Unit Specifications	299	2000
Attributes	299	2000
Area Fill	299	2000
Angle Display	299	2000
Partial Deletes	299	2000
Length Measurement	299	2000
Fillets	299	2000
Chamfers	299	2000
Mirroring	299	2000
Object Snap	299	2000
Curve Fitting	299	2500
Isometric Views	299	2000
Multiple Colors in Layer	299	N/A
Multiple Line Types in Layer	299	N/A
Single Point Move	299	N/A
High Res. Printer Support for Over 100 Printers	299	N/A
Merge Drawings	299	N/A
Single Keystroke Commands	299	N/A
HPGL Input	299	N/A
IGES Output	299	N/A
DEMO DISK	FREE	

PRODESIGN II — ONLY $299!

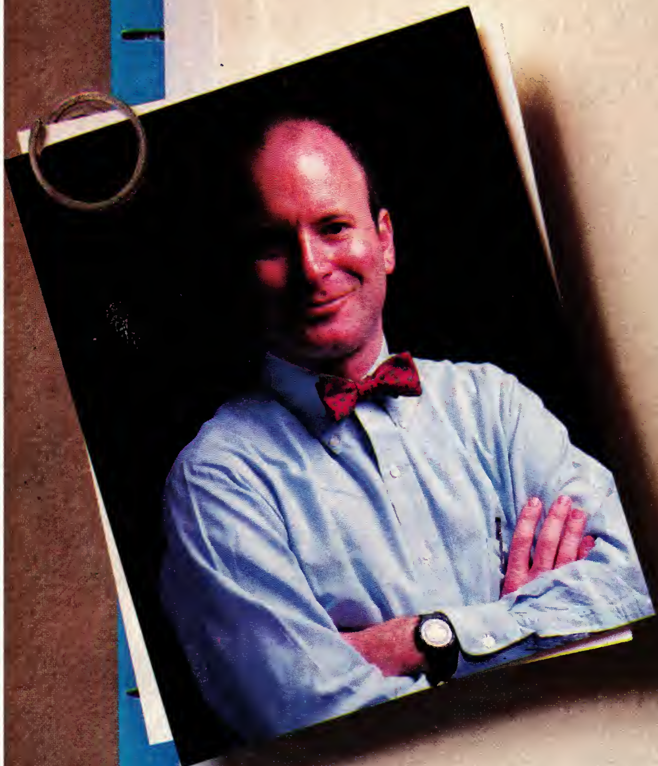


CIRCLE NO. 218 ON READER SERVICE CARD

```
        $D0/$E3/         { SHL BL,1          left 4 bits  }
        $25/$00/$0F/     { AND AH,0FH      discard old bg  }
        $00/$E3/         { ADD BL,AH       assemble color  }
        $B8/$DE/$09/     { MOV AX,09DE     chr ß to AH     }
        $B9/$01/$00/     { MOV CX,01       one to write    }
        $CD/$10);        { INT 10H         write chr, attr }
end;

{------- DETERMINE NUMBER OF ITERATIONS AT (x,y) -----------------}

Function Iterate(x,y:real):integer;
var
   scratch : integer;
begin
   Inline(
     $B9/$3F/$00/     { MOV   CX,3FH             # iterates -> CX }
     $9B/$D9/$E8/     { FLD1                     1 to 8087 Stack  }
     $9B/$D8/$C0/     { FADD  ST(0),ST(0)        2 on Stack       }
     $9B/$D8/$C0/     { FADD  ST(0),ST(0)        4 on Stack       }
     $9B/$DD/$46/$0C/ { FLD   QWORD PTR [BP+12]  x to Stack       }
     $9B/$DD/$46/$04/ { FLD   QWORD PTR [BP+4]   y to Stack       }
     $9B/$D9/$C1/     { FLD   ST(1)              Copy x           }
     $9B/$D9/$C1/     { FLD   ST(1)              Copy y           }
                      { HERE:                    Loop label       }
     $9B/$D9/$C1/     { FLD   ST(1)              Copy x           }
     $9B/$D8/$C8/     { FMUL  ST(0),ST(0)        x*x              }
     $9B/$D9/$C1/     { FLD   ST(1)              Copy y           }
     $9B/$D8/$C8/     { FMUL  ST(0),ST(0)        y*y              }
     $9B/$DE/$E9/     { FSUBP ST(1),ST(0)        x*x - y*y        }
     $9B/$D8/$C4/     { FADD  ST(0),ST(4)        x*x - y*y + x    }
     $9B/$D9/$CA/     { FXCH  ST(2)              new x <-> old x  }
     $9B/$DE/$C9/     { FMULP ST(1),ST(0)        x*y              }
     $9B/$D8/$C0/     { FADD  ST(0),ST(0)        2*x*y            }
     $9B/$D8/$C2/     { FADD  ST(0),ST(2)        2*x*y + y        }
     $9B/$D9/$C1/     { FLD   ST(1)              Copy x           }
     $9B/$D8/$C8/     { FMUL  ST(0),ST(0)        x*x              }
     $9B/$D9/$C1/     { FLD   ST(1)              Copy y           }
     $9B/$D8/$C8/     { FMUL  ST(0),ST(0)        y*y              }
     $9B/$DE/$C1/     { FADDP ST(1),ST(0)        x*x + y*y        }
     $9B/$D8/$DD/     { FCOMP ST(5)              Greater than 4?  }
     $9B/$DD/$7E/$FC/ { FSTSW [BP-4]             Status to Scratch}
     $9B/             { FWAIT                    8087 Done?       }
     $8A/$66/$FD/     { MOV   AH,[BP-3]          Status to AH     }
     $9E/             { SAHF                     Status to Flags  }
     $77/$02/         { JA    QUIT               x*x+y*y > 4?     }
     $E2/$C3/         { LOOP  HERE               No then Loop     }
                      { QUIT:                    Yes              }
     $89/$4E/$14);    { MOV   [BP+20],CX         Return # iterates}
end;

{------- MAIN PROGRAM BEGINS ------------------------------------}

begin
  LoRes;                                  { Switch to LoRes mode      }
  dx := xrange/159; dy := yrange/99;      { Scale world to screen     }
  y := ymin + yrange;                     { Maximum y to top of screen }
  for j := 0 to 99 do                     { 100 rows on LoRes screen  }
  begin
     x := xmin;                           { Minimum x to left of screen}
     for k := 0 to 159 do                 { 160 columns on LoRes screen}
     begin
        n := Iterate(x,y);             { Determine number of iterations }
        c := n div 8;                  { Determine color number 0..7   }
        if n mod 8 > 3 then c := c+8;
                                       { If remainder = 4..7 then bright }
        Point(k,j,c);                  { Plot point on screen          }
        if keypressed then goto quit;
                                       { Press any key to interrupt/quit }
        x := x + dx;                   { Update x coordinate of point  }
     end;                              { Loop until finished with row  }
     y := y - dy;                      { Update y coordinate of point  }
  end;                                 { Loop until finished with screen }
quit:  repeat until keypressed;        { Hold picture until key pressed }
  TextScreen;                          { Restore normal text screen    }
end.
```

C O M P A N Y   M I S S I O N

His mission is to streamline communications by making different brands of micros talk to each other in the same network.



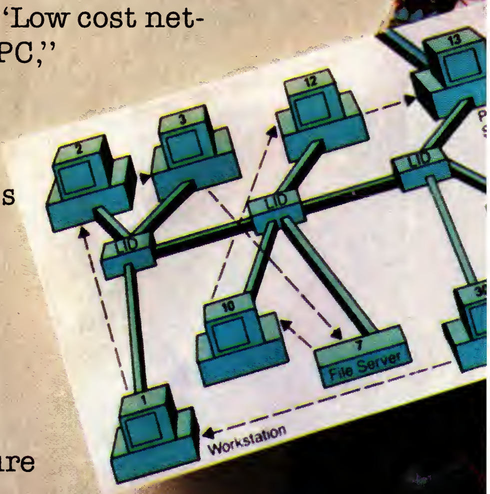
He's just one of 100,000 PC systems experts who uses PC Tech Journal to fulfill his professional responsibilities as a problem-solver. His recommendation is more often than not the last word in his company's computer purchases.

When one person has this much influence, he has to know more about state-of-the-art PC technology than anyone else in his company. That's why he reads PC Tech Journal.

Clues that put him on the right path

The articles in PC Tech Journal act as a compass to direct PC systems experts through the thickening forest of PC products.

This PC expert discovers that Bruce Churchill's articles, "Low cost networking for the IBM PC," points to many of the networking products he will need. And William Colsher's tips on sharing files keep him from wandering into the valley of lost data.

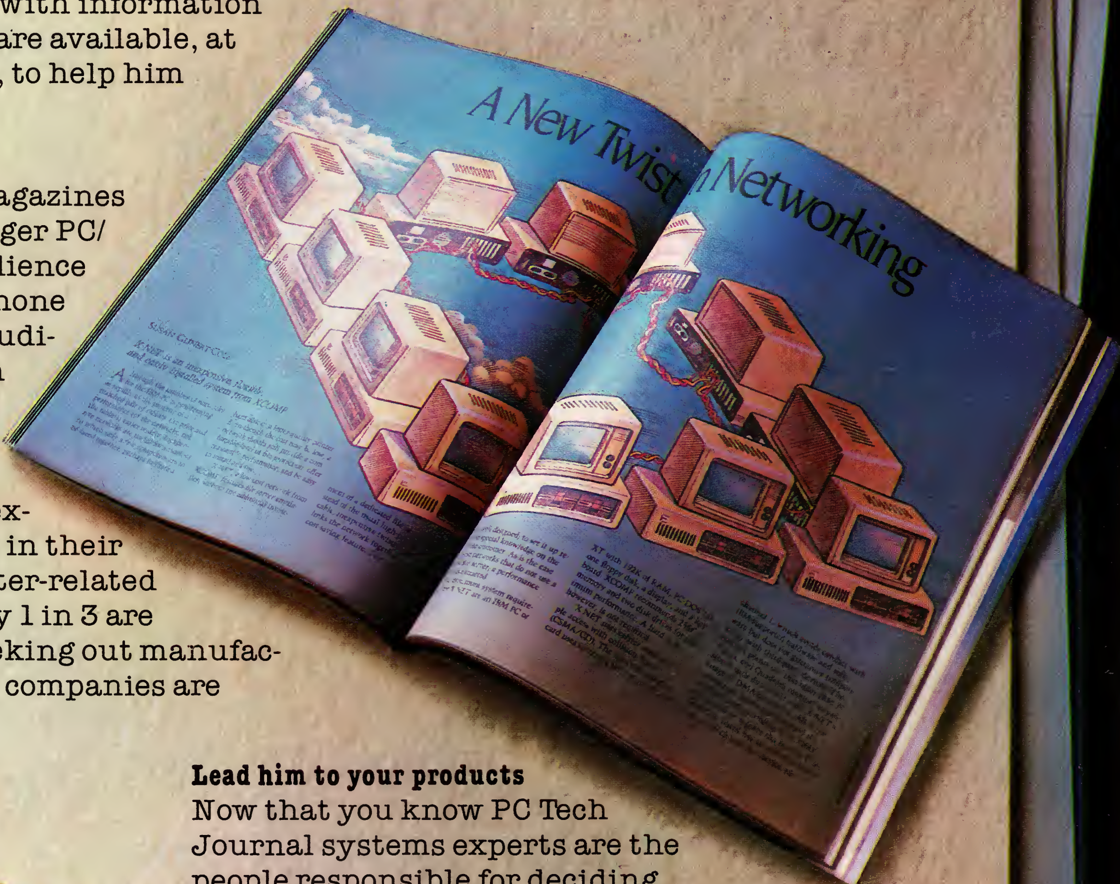


Clue that help him perform his best

It's not only the feature articles that give him direction, though.

C A S E 2 L O C A L A R E A N E T W O R K

The ads, too, keep him on course. They provide him with information on what products are available, at this very moment, to help him complete his task.



Stay hot on his trail

There are other magazines that give you a larger PC/PC compatible audience of IBM users, but none that give you an audience with so much buying power.

Eight out of ten PC Tech Journal systems experts are involved in their company's computer-related purchases. (Nearly 1 in 3 are responsible for seeking out manufacturers when their companies are acquiring LANs).

Lead him to your products

Now that you know PC Tech Journal systems experts are the people responsible for deciding between your product and a competitor's, reach them before the next decision is made.

Make it your mission to influence the buyers who influence their companies. It's easy! Just call your PC Tech Journal sales representative, now, and make PC Tech Journal his guide to your products.

PC TECH JOURNAL

Unmatched power in your market

Ziff-Davis Publishing, One Park Avenue, New York, NY 10016 (212) 503-5100

Focus on the technical issues that concern *you* in PC TECH JOURNAL

Let's face it, every day brings new technical challenges with the IBM PC. And just when you think you've met them all, you'll find out that IBM has just introduced DOS 3.1—or the new networking standard is likely to be the IBM token ring.

Where can you turn for accurate and in-depth answers to the technical questions that confront you daily? PC TECH JOURNAL.

PC TECH JOURNAL focuses on the technical what-if's and how-to's that are crucial to the growth of PC systems.

PC TECH JOURNAL will give you a clear picture of products and technical developments with detailed, objective articles on data management, applications development, communications, and programming. PLUS, you'll receive hard-nosed product evaluations of both hardware and software.

Your subscription to PC TECH JOURNAL includes the special PC TECH JOURNAL Directory issue, a comprehensive guide to IBM/MS DOS products and index to PC TECH JOURNAL articles!

Subscribe now and save 50% on the magazine that puts technical issues into focus. PC TECH JOURNAL. One year (13 issues) only $26.70. To subscribe, please use the attached Order Card, or call 1-800-852-5200 for fast service.
Savings based on annual single-copy price of $53.35.

Consult with the Experts.

TECH PC JOURNAL

4Z387

CIRCLE 353 ON READER SERVICE CARD

Macros provide a powerful programming tool when working with Lotus 1-2-3. However, Lotus's macros are difficult to write and document. QUICKCODE for 1-2-3 from Fox and Geller, Inc., designed for use with Lotus 1-2-3, makes programming macros easier.

QUICKCODE for 1-2-3 requires Lotus 1-2-3 version 2, DOS 2.0 or later, and 384KB. The program is not copy protected. After QUICKCODE for 1-2-3 has been loaded into memory, it automatically loads Lotus. When the user exits Lotus, QUICKCODE for 1-2-3 automatically removes itself from memory. The program reduces the space available for worksheets by about 126KB.

The QUICKCODE program includes three components. The most impressive is the macro language PL/123, which is similar to BASIC and dBASE languages. The other two components are a keystroke recording capability, which allows the user to record 999 keystrokes, and a facility for developing databases and supporting macros.

ing commands. The window limits programs to 300 lines. Because the code produces ASCII files, any word processor can be used.

The facility for recording and playing back keystrokes is the easiest of the QUICKCODE for 1-2-3 functions to use. This facility is a standard keyboard enhancer with the added ability to produce Lotus macros directly. Unfortunately, it is poorly designed and incompatible with SideKick. Although the manual claims that QUICKCODE can store 1,000 keystrokes, the program stops recording after 999 and displays a message stating the buffer has 1 percent of memory remaining. The program allows only one series of keystrokes to be stored in memory at a time. If the user begins to record a second series, the program erases any existing keystrokes without warning. Keystrokes stored in memory can be written to a file or to the current worksheet as a macro.

The user interface has several rough edges. A QUICKCODE prompt reminds the user that Ctrl-S can be used to end the recording of keystrokes, but the prompt disappears when

also can contain field length, type, default value, and error checking. The manual claims the program can create a database with as many as 204 fields; however, during testing, only 86 fields were possible. Trying to produce a database with a greater number of fields resulted in error 244, which is not mentioned in the manual.

After the specification file is created, a database generation program is run from DOS. The final step in creating a database involves running a third program from Lotus with QUICKCODE in memory. A three-field database created with QUICKCODE during testing required 16KB of memory.

A QUICKCODE database provides a data entry mode and a data viewing mode. The program permits viewing of only one record at a time. The data-entry screen does not display the command that signals the end of data entry. Further, QUICKCODE databases cannot be easily converted to stand-alone databases; the user must be familiar with QUICKCODE commands.

While testing for this review, a number of operations caused QUICK-

CODE to hang the computer. In most instances the problem was the result of incompatibility between QUICKCODE and memory-resident programs, such as SideKick or SuperKey. For example, pressing the help key (F1) more than one time and pressing Ctrl-Break to end a macro, then using the pop-up window to edit a macro program both caused the computer to hang. Trying to save an empty window and pressing more than one key at a time while editing in the window also hung the system.

Pressing Ctrl-Break to end a macro, then using the pop-up window to edit a macro program also hung the system. This problem, however, did not seem to result from QUICKCODE's incompatibility with memory-resident programs.

QUICKCODE's manual is paperback, which is difficult to keep open while working through the tutorial. The installation section of the manual has many minor errors, as do the examples in the tutorial. For example, one program divides by 10, but the text explains it is dividing by 100. Several programs use a counter, but it is not set at the beginning of the program. The resulting macro runs once because the cell used to store the counter value is empty and has a value of 0. The macro will not run again, however, because, after the first execution, the cell contains the value used to stop the macro. The typeset manual makes it difficult to discern spaces, and in some programs (especially Lotus macros) spaces are critical. With the exception of a single explanation line below each menu choice (similar to menus in Lotus 1-2-3), QUICKCODE provides no help facilities.

The technical support from Fox and Geller, Inc. stated that QUICKCODE users can sell the macros and databases they create without making royalty payments to the company. The manual does not mention this, however.

Fox and Geller, Inc. also makes a QUICKCODE for the dBASE family of products, which is available for $295. (For a review of the dBASE version, see "Instant Gratification: QUICKCODE," Susan Glinert-Cole, June 1984, p. 129.)

The current release of QUICKCODE for 1-2-3 seems to be a program that was rushed too quickly to market. It cannot be recommended as a keyboard macro program or as a tool for developing databases. However, PL/123 is a valuable tool for writing macros. It is much easier to use and to read than Lotus 1-2-3's native macro language.

—RONNY RICHARDSON

LET'S C

Mark Williams
1430 W. Wrightwood Avenue,
Chicago, IL 60614
312/472-6659

PRICE: $75

CIRCLE 364 ON READER SERVICE CARD

Let's C from Mark Williams is a smaller and less expensive version of the C Programming System for the PC. However, the Let's C compiler is surprisingly similar to its more expensive counterpart. Features of the Let's C compiler can be compared to those of the C compilers that were reviewed in "The State of C" (William J. Hunt, January 1986, p. 82).

Let's C is intended for the casual or small-project user. It supports only small model programs (64KB of code and 64KB of data). Users interested in writing larger programs can upgrade to the C Programming System compiler and receive a discount equal to the original purchase price of Let's C.

The basic specifications for Let's C are presented in table 1; table 2 provides an outline of the compiler's features. With the exception of the restricted program size, Let's C can be considered a full-featured compiler—it provides full support of the C language (including register variable optimization) as well as several extensions (such as void functions, enumerated types, and structure assignments). Let's C is sufficiently compact to fit easily on a dual-floppy-disk system, and the package includes several convenient UNIX-like utilities, such as **egrep**, **pr**, and **wc**, along with the compiler, linker, and librarian. Like most C compilers in this price range, Let's C includes a small, full-screen editor (a miniature version of the EMACS editor from UNIX) for users who may not want to invest in a full programmer's editor for their system. The editor is provided in source form for customizing and study.

TABLE 1: *Specifications*

Version tested	3.02
Supported on other systems	See text
Cross-compiler hosts	See text
Availability of libraries	Fair
Minimum disk space required	330KB
Minimum RAM	256KB
Supports full language	Yes
Full standard library	Yes
PC-specific functions	Yes
Assembly language interface	Yes
COMPATIBILITY	
MASM	—
LINK	—
SOURCE CODE	
Start-up sequence	—
Library functions	—
MEMORY MODELS	
Large	—
Medium	—
Compact	—
Small	Yes
.COM	—
OTHER PROGRAMS INCLUDED	
Librarian	Yes
Asssembler	Yes
Linker	Yes
Source-level debugger	Separate
MAKE	—
Other	GREP, PR, Edit., etc.

These specifications can be compared with those for other C compilers in table 1 in "The State of C" (William J. Hunt, January 1986, p. 84).

The Let's C compiler supports the small memory model only. It is not compatible with MASM, and the UNIX-like assembler included with Let's C is not as powerful as MASM.

Documentation for Let's C consists of a single, spiral-bound manual—the same manual included with the C Programming System. The sections applicable only to the larger compiler are indicated with shaded text. Overall, the quality of the documentation is good; it explains clearly the compiler's operation and options. The section dealing with the assembler and linker, however, is weak. See table 3 for an overall rating of the compiler's documentation.

Installation of the Let's C compiler also is clearly explained in the documentation. A separate directory can be used to hold the compiler and linker, the libraries, and the **#include** files. Operation of the system is convenient. The package includes compilation options that allow the programmer to

TABLE 2: *Compiler Features*

COMPILER OPERATION	
Single-step compile command	Yes
Compile and link	Yes
Accepts lists of files	Yes
Accepts wild cards	Yes
Lists preprocessor output	Yes
Lists assembler output	Yes
Line numbers in error messages	Yes
Header file search list	Yes
Flexible disk file layout	Yes
C LANGUAGE EXTENSIONS	
Embedded assembly language	—
Void function returns	Yes
Enumerated types	Yes
Structure assignment, etc.	Yes
Function argument checking	—
LIBRARY EXTENSIONS	
Math functions (sqrt, exp, etc.)	Yes
Unbuffered file I/O	Yes
Keyboard input (low-level)	Yes
PC screen output (cursor control, cursor attributes, scroll)	Some
Execute programs/DOS (exec/fork and system)	Yes
MS-DOS services (date, time, etc.)	Some
PC-specific functions	Some
UNIX-compatible functions	Yes
Error recovery (setjmp( ), longjmp( ))	Yes
FILE I/O	
Redirection	Yes
Full path names	Yes
DOS 1.1 support	—
DOS 3.1 file sharing	—
Record locking	—
ASCII/binary mode	Yes
MEMORY USAGE	
Overlays	Yes
Default stack size	Yes
Stack size settable	Yes
Stack overflow checking	—
8086 FAMILY SUPPORT	
Byte/word alignment	—
80186/80286 support	—
8087/80287 support	—
Automatic sensing	—
ROM support	—

These compiler features can be compared with those for other C compilers in table 2 in "The State of C" (William J. Hunt, January 1986, p. 86).

Let's C is easy to install, and the compiler options are well documented. It is as full featured as many $500 products.

TABLE 3: *Documentation Quality*

INSTALLATION	
Packing list	Yes
File inventory	Yes
Key files described	Yes
Quick step-by-step procedure	Yes
Instructions for disk configurations	Yes
List changes from last version	—
SET-UP	
Set-up assumptions described	Good
Notes on RAM/second hard disk	Poor
OPERATIONS EXPLAINED	
Compile options	Good
Compiler error messages	Good
Linking C programs	Fair
Runtime error messages	Good
Runtime options	Good
LANGUAGE/LIBRARY SPECIFICATIONS	
Deviations from Kernighan and Ritchie standard	Good
Data type representation	Good
Memory models and memory layout	Fair
DOS and PC-specific features	Fair
ASSEMBLY LANGUAGE INTERFACE	
Segment, group, and class specification	Fair
Standard prologue, epilogue	Poor
Instruction formats for args, public, extern, struct	Poor
Return value conventions	Fair
Complete examples	Poor
FILE I/O	
Redirection	Fair
Console I/O	Good
Device I/O	Good
Buffered versus unbuffered	Good
ASCII versus binary modes	Good
LIBRARY DOCUMENTATION	
Average lines per function	20
Cross-reference information	Good
Functions in table of contents	Good
Examples of use	Poor
MANUAL ORGANIZATION	
Detailed table of contents	Good
Index with functional entries	Fair
Order of function documentation	Alpha.
OVERALL RATING	Good

These notes on the documentation for the Let's C compiler can be compared with those for other compilers in table 3 in "The State of C" (William J. Hunt, January 1986, p. 88).

The documentation supplied with Let's C is the same as that included with the C Developer's System.

check programs rigorously for compliance with the C language as described in Kernighan and Ritchie's *C Programming Language* (Prentice-Hall, 1978). Thus, casual or infrequent C programmers are able to write clean programs with a minimum of trouble. Both the compiler and the linker are reasonably fast when used for development; overall, the linker is slightly slower than LINK (see table 4 for benchmark times).

Let's C outputs only Mark Williams' proprietary object module format. In addition, output produced by the assembler is readable only by the UNIX-like assembler included with the package, not by MASM. This may not be a problem for users content with rewriting only a few functions in assembly language to improve program execution speed. For users working with larger projects, however, this restriction can prove a real nuisance. Mark Williams' object format is supported by some (not all) add-on library vendors, including Greenleaf and C Power Packs from Software Horizons, Inc.

TABLE 4: *Benchmarks*

COMPILE TIMES	
60-line file	20.9
150-line file	37.6
500-line file	60.6
LINK TIMES	
1 object file	35.8
6 object files	41.4
PROGRAM SIZES (bytes)	
Eratosthenes sieve	12,183
Pentathlon	15,480
GENERAL OPERATIONS	
Function calls (Fibonacci)	28.7
Integer arithmetic	36.9
Long arithmetic	199.8
Subscripts (character count)	28.0
Pointer use (string copy)	37.6
With register variables	31.3
Eratosthenes sieve	24.2
With register variables	18.7
FILE I/O	
Read/write	
Floppy disk to floppy disk	19.5
Hard disk to hard disk	5.0
Getc/putc	
Floppy disk to floppy disk	9.1
Hard disk to hard disk	5.3
Floating-point operations	
Add/multiply (dot product)	30.0
Exp/log	34.9
Sin/tan (trig functions)	41.4

Times are shown for the small model only.

Benchmarks were run on an Alpha Micro Workstation, the programs were run from a hard disk. This should be considered when comparing these benchmark times to those in table 4 in "The State of C" (William J. Hunt, January 1986, p. 90), which were reported for a PC/XT.

These benchmark times indicate that Let's C is simply the C Developer's System without the extra memory models and MASM compatibility.

The benchmarks shown in table 4 were run on an Alpha Micro Workstation (a PC/XT clone) equipped with 512KB of memory, a 10MB hard disk, and an STB Chauffeur video board. CONFIG.SYS was present with BUFFERS=20 and FILES=20. The results show that the .EXE files produced by the compiler are smaller than average. In addition, with the exception of the Sieve, execution times were somewhat slower than average, and the result of the long-arithmetic benchmark was much slower than average. Using register variables improves performance slightly. The feature and benchmark tables (see tables 2 and 4, respectively) indicate that Let's C is, in short, the C Programming System without the memory models or MASM compatibility.

The performance and documentation of the $75 Let's C compiler rival those of C compilers for the PC currently being sold for $500. Let's C represents a professional-quality compiler at a bargain price.

In addition, the Let's C compiler, like the full C Programming System, can be used with the Mark Williams **csd** source debugger. While this debugger suffers from some limitations (for example, it is unable debug assembly language or library functions) and some irritations (especially its habit of proofing input key by key rather than when Return is pressed), it is generally considered adequate for use with small programs. Like Let's C, the debugger is priced at $75; thus, a complete C programming system that is sufficient for use with small projects can be purchased from Mark Williams for only $150.

Let's C goes a long way toward bringing the power of C to the average

PC programmer at an affordable price. It is not intended for users requiring a large memory model attached to an assembler with the power of MASM. However, this compiler can be highly recommended to users who require an excellent small model implementation with a quick compile-link-run cycle.

—MARTY FRANZ

GENSCREEN
Personal Computer Development Corporation
P.O. Box 8556, Warwick, RI 02888
401/333-8704

PRICE: $69.99

CIRCLE 354 ON READER SERVICE CARD

COBOL developers may find a time saver in GENSCREEN, a utility that generates screens for use in applications. Available from Personal Computer Development Corporation, the product is designed for use with programs written in Microsoft COBOL version 2.0 or later and IBM COBOL version 1.0. (According to Personal Computer Development Corporation, GENSCREEN is the only screen generator available for use with Microsoft COBOL that produces compilable source code.)

The program's objective is to reduce the time required to develop and maintain the screen section of a Microsoft COBOL program. To this end, it takes over the tedious task of generating the data statements needed to support screen input and output. GENSCREEN reads an ASCII line sequential file that contains an image of the screen, which can be created with any program editor or word processor capable of outputting a standard ASCII file. In addition to the screen's labels, the screen file also contains delimiters that are used to define the beginning and end of each input or output field.

Unlike some other screen utilities, GENSCREEN does not include a screen painter; the developer must depend on a program editor to create screen images. This can be an advantage when designing a screen because text editors are more powerful and often easier to use than screen painters. On the other hand, if the user's text editor does not support input of non-keyboard characters via the Alt key and numeric keypad, the user will not be able to include graphics characters in screen designs.

After a screen image is created, the user can choose to allow GENSCREEN to generate data names for input fields to create a second file containing a list of custom data names.

GENSCREEN presents a display that allows users to enter parameters for the names of the input and output files, for a fill character for alphanumeric fields, and for the default screen colors. This display is supported by pop-up help windows that almost eliminate the need to refer to the documentation.

GENSCREEN's strength lies in its ease of use. The documentation is short but complete. During testing, the pamphlet was read, the software installed, and a sample screen generated—all in less than 30 minutes. A menu screen with borders and icons was created in an additional 20 minutes.

The penalty for GENSCREEN's ease of use is limited features. GENSCREEN can handle the screen only as a whole—with the exception that it marks the input fields in the screen image. All fields in the GENSCREEN-generated screen section are assigned the same default attributes. As a result, the user cannot take advantage of the complete set of attributes provided by the compilers the product supports. Additional features can be added by editing the copy files produced by GENSCREEN. Any changes to the copy files must be repeated if the screen layout is regenerated.

Two points must be remembered with GENSCREEN. First, the screen image created with the text editor is limited to 80 columns and 25 lines. Second, a space must follow the terminating delimiter of each input or output field.

As a result of the second stipulation, three blank spaces always separate any two fields on the same display line; if borders are used, at least two spaces separate the fields and the screen border. Thus, GENSCREEN cannot convert screens generated in some other COBOL implementations to Microsoft COBOL. However, the product did prove to be a time saver in most cases, even though some editing was required to produce custom screens.

GENSCREEN is also limited in its handling of screen attributes. Only the foreground and background color of the entire screen can be set within GENSCREEN. To change the color (or any other attribute) of a specific field, the user must manipulate the COBOL code produced by GENSCREEN.

GENSCREEN performed well during testing. The only problem surfaced when an editor was used that embedded Tab characters in the screen image file. GENSCREEN did not process the Tab characters as five spaces. The Personal Computer Development Corporation assures this problem will be corrected.

GENSCREEN is not a screen manager and does not reduce the coding effort necessary for handling screen edits and validations. Nonetheless, the product can save time. It is best suited for programs that will remain fairly static.

—JOHN C. NOLAND

PC ⇔ MAINFRAME
VIA 9-TRACK TAPE

For Information Interchange – Backup – Archival Storage

IBM format compatible 9-track, ½ inch magnetic tape is the universally accepted media for mainframes and minicomputers. Catamount offers *Low Cost, Lightweight* 9-track Tape Subsystems for the IBM-PC/XT/AT computers which allow:

- Reading tapes generated on mainframes and minicomputers.
- Writing tapes to be read on mainframes and minis.
- ASCII, EBCDIC and Binary tapes accommodated.
- 800 bpi NRZI, 1600/3200 bpi PE, and 6250 bpi GCR format systems available.
- Storage capacities up to 270 MB on a single reel.

Systems come complete with comprehensive DOS command syntax oriented software and an Installable Device Driver. For OEM applications, the tape controller is available separately.

CATAMOUNT CORPORATION
2243 Agate Court
Simi Valley, CA 93065-1898
(805) 584-2233

CIRCLE NO. 111 ON READER SERVICE CARD

Programming

SR. SOFTWARE ENGINEERS

ASHTON-TATE is one of the nation's most dynamic leaders in the microcomputer software industry, publishers of dBASE III Plus ®, Framework II® and MultiMate®. We currently have a challenging opportunity for a few qualified Senior Software Engineers in our Product Development Centers.

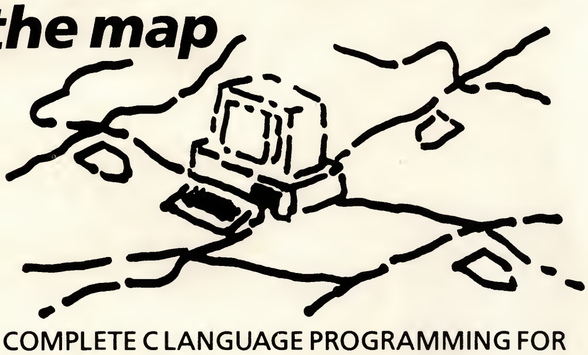
Selected candidates for these key positions will be senior designers of systems software, database internals, and/or user interfaces; and have a working knowledge of "C" and MS-DOS. Experience with ASHTON-TATE products, database applications and UNIX a plus.

ASHTON-TATE offers a competitive salary and comprehensive benefits package. For immediate consideration, send your resume to: **Mr. Tony Faison, ASHTON-TATE, 20101 Hamilton, Torrance, CA 90502.** Equal Opportunity Employer.

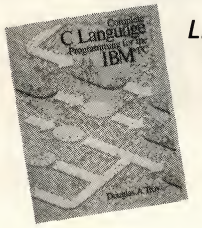
ASHTON·TATE

CIRCLE NO. 134 ON READER SERVICE CARD

Our C-language book puts your IBM® PC on the map



COMPLETE C LANGUAGE PROGRAMMING FOR THE IBM® PC by Douglas A. Troy combines the two most popular computing tools on the scene today. Using syntax diagrams—literally the "road maps" of the language—this step-by-step guide will steer you all the way from theory to applications.



Little, Brown and Company
Boston, MA 02108

400 pages $24.95
ISBN 117943

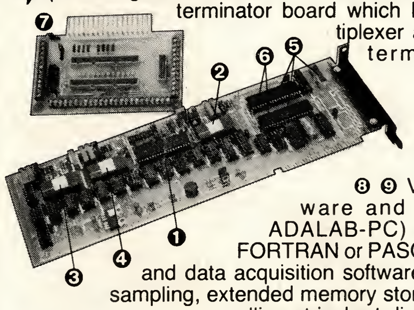
To order, call
800-343-9204

CIRCLE NO. 173 ON READER SERVICE CARD

ADALAB-PC™
Adds Lab Data Acquisition To Your IBM®PC For $655

An ideal interface for chromatography, spectrophotometry, process control, etc.

ADALAB-PC includes ❶ a 13-bit integrating A/D, ❷ a 12-bit fast A/D (up to 20 kHz, add $275), ❸ a 12-bit D/A voltage output, ❹ a second 12-bit D/A (add $55), ❺ 32 digital I/O bits, ❻ four 16-bit timers or counters, and ❼ a terminator board which has a 4-channel multiplexer and convenient screw terminals (16-channel differential analog multiplexer with programmable gains from 1 to 256, add $300).

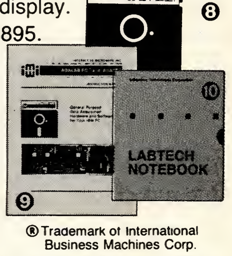


❽ ❾ Versatile ADAPT software and manual (free with ADALAB-PC) works with BASIC, FORTRAN or PASCAL, includes self-test and data acquisition software. Supports fast DMA sampling, extended memory storage, scrolling stripchart display.

❿ LABTECH NOTEBOOK software $895.

⓫ IMI is an IBM Value Added Dealer.

COMPLETE ADALAB-PC SYSTEMS at $3595, including 256K IBM PC, 360KB disk, PC DOS 3.0, green monitor, color/graphics adapter, Epson printer with adapter & cable, ADALAB-PC, ADAPT software, plus IMI's Scientific Plotter-PC ($95) and Curve Fitter-PC software ($95) *FREE* with systems!



® Trademark of International Business Machines Corp.

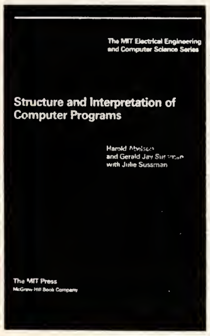
INTERACTIVE MICROWARE, INC.
P.O. Box 139, Dept. 237 Telex: 705250
State College, PA 16804 (814) 238-8294

CIRCLE NO. 209 ON READER SERVICE CARD

An Elegant Scheme

An advanced introduction to the principles of computer science, this text is also an invaluable workbook for the Scheme programmer.

Structure and Interpretation of Computer Programs
Harold Abelson and Gerald Jay Sussman with Julie Sussman (MIT Press/ McGraw-Hill, Cambridge, MA 1985)
542 pages, hardcover, $34.95



Structure and Interpretation of Computer Programs was developed from the course notes for an accelerated introductory computer science course at Massachusetts Institute of Technology (MIT). Written by Professors Harold Abelson and Gerald Jay Sussman, it covers subjects ranging from procedural abstraction to compiler implementation, from data-directed programming to register machines—all in a style that is universally praised by students.

This book is designed for readers with a working knowledge of the programming language Scheme, a dialect of LISP developed at MIT. As a text, it is not for casual reading and is not an introduction to the language. Rather, it is a complete introduction to the fundamental principles of computer science, and in this it leaves no stone unturned.

Scheme is a lexically scoped, interpreter-based language in which procedures are treated as first class data objects, so that programs may be manipulated as data. Its expressive power and simple syntax make Scheme ideal for illustrating the fundamental concepts of computer science. All the programming exercises and examples in the text are written in Scheme.

The introductory chapter begins with a brief explanation of Scheme which should enable the reader to write simple programs. Features and variations of the language are introduced in later chapters.

Explanations of iterative and recursive computation, orders of growth, and the passing of procedures as parameters and results follow the introductory section on Scheme. The latter capability is one of Scheme's most powerful and makes a whole host of data and control abstraction mechanisms possible.

Chapter 2 explains the use and implementation of data types, using sequences, lists, trees, sets, and real, complex and rational numbers as examples. The text emphasizes simple, functional implementations of data abstractions and provides code for a symbolic differentiator, a Huffman-encoder, and a generic arithmetic package for illustration. Explanations of data-directed programming and message-passing and example programs from earlier in the chapter are rewritten using these techniques.

In chapter 3, the authors provide a fresh perspective on the eternal problem of programming side effects. They deal with issues of local state, including assignment and mutable data objects and the costs and benefits thereof.

This chapter also describes the stream, which is a programming model analogous to the electrical engineer's signal-processing model for circuits. A full implementation of streams, written in Scheme, is presented as a practical means by which to avoid the problems that are usually associated with assignment and mutable data.

The application of computer languages as an abstraction mechanism, including a discussion of elementary language implementation, is covered in chapter 4, "Metalinguistic Abstraction." The authors provide a detailed analysis of the workings of the Scheme interpreter and include code that will implement both the Scheme interpreter and a logic-based database query language. Special attention is paid to binding disciplines (for example, lexical versus dynamic scoping) and the environment model of evaluation.

Abelson and Sussman use a simple register-machine simulator to explain assembly-language programming concepts, including: memory and addressing, registers, flow of control, stack operations, subroutine calls, and tail recursion. After the simulator has been introduced and thoroughly exercised, the authors discuss compilers and their implementation. The text includes a complete compiler, written in Scheme, to convert Scheme expressions to corresponding register-matching code.

Challenging, thought-provoking exercises appear throughout the text. In many, the reader is asked to add or modify code to complete or change the behavior of sample systems. Imaginary characters Ben Bitdiddle and Alyssa P. Hacker pose problems and entertain the reader with their exploits.

All sample code in the text will run without change on Texas Instruments' PC Scheme for the TI Professional and IBM PC. *Structure and Interpretation of Computer Programs* was written with a hands-on learning environment in mind, and PC Scheme complements it well. (See Product of the Month in this issue, p. 29, for more details.)

While it is not light reading, *Structure and Interpretation of Computer Programs* is clear, concise, and frequently amusing. Its emphasis on simple, elegant solutions to problems, and the wide variety of programming tools and techniques it presents, make it a valuable reference book.

—ARTHUR GLECKLER

FREE CATALOG AVAILABLE

The U.S. Government Printing Office offers a free catalog of 1,000 of the government's best sellers. The *US Government BOOKS* catalog includes nearly 20 titles on computer science. It can be ordered by writing to: New Catalog, P.O. Box 37000, Washington, DC 20013.

—*WF*

New for IBM PC/XT/AT low cost mag tape subsystem

- Fast controller—40Kbytes/second
- New MAINSTREAMER™ tape drive
- Small size, lightweight

Send for our complete drive and interface manuals so you can evaluate and compare. Only $25 for both.

Tape systems are also available for all other RS-232-C interfaces. Call or write attn: IBEX Applications Engineer today.

IBEX *Right for the times*
IBEX COMPUTER CORPORATION
20741 Marilla St., Chatsworth, CA 91311
(818) 709-8100 TWX: 910-493-2071

CIRCLE NO. 149 ON READER SERVICE CARD

OVERCOME
FORTRAN | PASCAL | C
LIMITATIONS WITH

$129 NO LICENSE FEE

NO LIMIT

Ver. 3.0 For: MS FORTRAN/Pascal/C IBM Professional and R-M FORTRAN SuperSoft FORTRAN

NOW SUPPORTS IBM ENHANCED GRAPHICS AND HERCULES MONOCHROME GRAPHICS

A library of over 90 Assembler routines transform FORTRAN, Pascal and C language compilers into the flexible, responsive, complete languages needed for the microcomputer environment. Hundreds of NO LIMIT owners are creating highly interactive software systems, often utilizing existing mainframe code and saving time and money in the process. Ver. 3.0 features include:

EXTENSIVE GRAPHICS	(Get, Put, Paint, Color, Dot, Line, Box, Circle, Ellipse)
FULL SCREEN CONTROL	(Windows, Cursor, Read/Write Screen)
STRING MANIPULATION	(Match, Compare, Concatenate/Extract, Pack, Justify)
KEYBOARD CONTROL	(Read Key During Execution, String Read With Edit)
FILE MANAGEMENT	(Exist?, Rename, Delete)
COMMUNICATIONS	(Full Interrupt Driven to 9600 Baud, Set Com Line, Send/Receive, Line/Modem Status)
INTERRUPTS	DOS Execution, Program Execution
OTHER FEATURES	(Command Line Read, DOS 3.1 Directories, Peek, Poke, Random Numbers, System Status, etc.)

And to complement NO LIMIT, the I/O PRO development system allows creation and editing of FORTRAN/Pascal/C callable screens. This word processor type system pays for itself several times over on your first project. $261.

M | E | F Environmental Inc.
P.O. Box 26537 Austin, Texas 78755
(800) 562-9700 (512) 251-5543 (Texas)

©Trademark of Microsoft Inc., Supersoft Inc., IBM Corporation, Ryan-McFarland and Microsoft Inc., respectively.

CIRCLE NO. 133 ON READER SERVICE CARD

dBASE II III to C

the dBx™ translator

- **dBx** produces quality **C** direct from dBASE II or III programs.
- Move dBASE programs to UNIX or other machines.
- Improve program speed and reliability.
- Support multi-user/network applications.
- With power guidebook of conversion hints.
- Includes full screen handler and uses your current **C** database manager.
- May be used to move existing programs or help dBASE programmers learn **C** easily.
- For MSDOS, PCDOS, UNIX, XENIX, Macintosh, AMIGA. (Uses ANSI.SYS driver on MSDOS, CURSES under UNIX)
- Priced from $350, also available from distributors.

dBx is a trademark of **Desktop Ai**

1720 Post Road E., Westport, CT 06880 MCIMAIL • DESKTOPAI
Phone • 203•255•3400 Telex • 6502972226MCI

CIRCLE NO. 102 ON READER SERVICE CARD

Does your database program let you...

1. Add and delete fields from existing files without destroying files?

2. Increase or reduce field width without destroying files?

3. Avail the updated data to all user files in the system?

4. Use the existing data instead of re-entering it as original data?

5. Create hierarchy and network databases easily?

6. Keep a true database, not just manage files?

TrueDataBase does! (and much more.)

For further information contact:
Software That Really Works, Inc.
750 E. Colorado Blvd. Ste 4
Pasadena, Ca. 91101
(818) 798-1661

CIRCLE NO. 147 ON READER SERVICE CARD

MAX STUL OPPENHEIMER

Taxing Software

A proposed regulation in California would single out software transactions for a special tax.

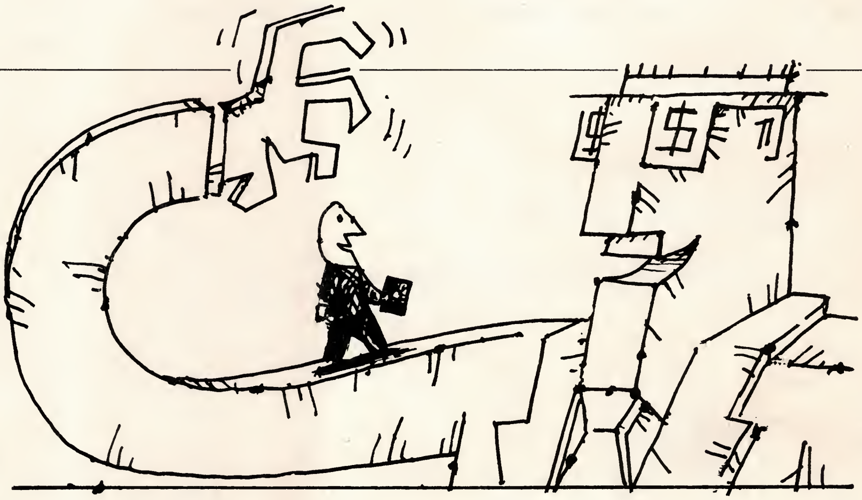


ILLUSTRATION • MACIEK ALBRECHT

Two-and-a-half years ago this column noted two approaches being taken to the taxability of the sale of software: some states were treating software encoded on disks the same as music encoded on records and therefore taxing the sale to the user (even in the case of custom programming); others viewed computer programming as a service and therefore not subject to sales tax. (See "Sales Tax and Software: Nothing is Simple," Legal Brief, Max Stul Oppenheimer, January 1984, p. 196.)

The issue is whether a program diskette is $1.50 worth of tangible property with some very valuable intangible (and therefore nontaxable) property encoded on it, or is it a very valuable piece of tangible property (and therefore taxable at the sales price). The issue could be left to the courts to resolve under general tax statutes, or it could be resolved definitively by specific legislation purposely directed at software transactions.

Taxation is a mixture of logic, fairness, and policy. If the United States Congress wants to reduce the use of gasoline, it can force the price up by adding a tax to the sale of the product. Likewise, if California has decided to kill off its software industry for public policy reasons, it can probably do so.

California's State Board of Equalization (SBE), which is charged with collecting sales tax and therefore with interpreting the sales tax statute, has proposed what seems an illogical method of applying sales tax to software.

The pattern of California's sales tax statute, greatly simplified, is as follows: retail sales of tangible personal property are taxable; sales to a retailer for resale are not; sales by the retailer to the consumer are. On the question of how to deal with computer programs (where intangible data is encoded on tangible media) the California legislature took the middle ground, applying sales tax to "canned" programs (those intended for general or repeated use) but not to "custom computer programs," defined as programs "prepared to the special order of the customer," including "those services represented by separately stated charges for modifications to an existing program which are prepared to the special order of the customer...Modification to an existing program to meet the customer's needs is custom computer programming only to the extent of the modification."

A specific legislative finding and declaration was made that "sales and service of custom computer programs...are service transactions not subject to sales or use taxes under any existing state law. The use of any storage media in the transfer of custom computer programs is only incidental to the true object of the transaction, which is the performance of a service."

The SBE has proposed Draft Regulation #1502, an interpretation of the statute. If the sale of a disk to a user may be taxed, why not tax the sale of the master disk from the author to the publisher? (Several answers are discussed below.) Try to decipher what the SBE is doing with these regulations:

1. The transfer of title of tangible personal property including property on which...information has been recorded...is a sale subject to tax.
2. Charges for modifying consumer-furnished tangible personal property (cards, tapes, disks, etc.), including charges for recording...information on...such tangible personal property are generally subject to tax.
3. Leases of tangible personal property may be taxable under certain conditions. In the case of canned programs the tax applies to license fees.
4. The sale or lease of a prewritten program is not taxable if the program is transferred by telecommunications from the seller's place of business to the purchaser's computer.
5. The tax does not apply to sales of custom computer programs, which include a program prepared to the special order of a customer who will use the program to produce and sell or lease copies of the program.
6. The transfer of a prewritten program on storage media is not a sale for resale when the storage media, or an exact copy, will be used to produce additional copies of the program.

The fifth and sixth provisions seem inconsistent, although they can be reconciled. One interpretation might be that, while the sale of a program to a publisher is not "a sale for resale," it is exempt from sales tax because it is a "special order." The problem with this interpretation is that it makes provision

number 6 nonfunctional unless the SBE intends to draw a distinction between a software author commissioned to write a program (sales tax exempt according to provision 5) and one who writes without a commission, then attempts to sell the program (taxable according to provision 6). This seems an illogical interpretation. Nonetheless, according to the SBE's principal tax auditor, is it precisely the regulation's intention.

The proposed regulation is inadvisable for several reasons. Fairness is the first. Under the typical sales tax statute in the United States, tax is imposed only on the sale to the end user. When property is sold from a manufacturer to a middleman for resale to the public, the sale to the middleman is tax exempt. Some countries impose a value-added tax on every sale, from the sale by the producer of raw materials to the sale to the end user. This tax applies only to the value added by each process. If $1 worth of trees are grown and then sold, a tax is paid on $1; when those trees are used to produce $10 worth of paper for sale, the applicable sales tax on the paper is based on $9; when $100 worth of books is printed on the paper, the sales tax is based on $90. The total tax is the tax rate multiplied by $100 (the end value to the consumer), but it is paid in installments. The proposal by the SBE would result in the total tax being paid on more than 100 percent of the end value to the consumer.

Why is the SBE singling out the software industry for a special tax? The California proposal treats print authors differently than software authors because the print publisher does not make an exact copy of the manuscript, but, according to the SBE, the software publisher does.

Industry opposition to the proposed regulation is gathering momentum. The Software Entrepreneur's Forum and the Software Services Association are fighting the proposal. (Both may be reached through Michael Odawa at P.O. Box 6413, San Jose, California 95150.) If the California SBE persists, and the state legislature does not step in, software authors and publishers in California will have to find ways of dealing with the new regulation. Following are some suggestions discussed in increasing order of cost (and increasing order of the probability of success).

1. Do not allow a publisher to sell an identical copy of your program. For example, supply the program as a print-out, charge a nominal sum for the print-out but a fortune for the right to transfer it to a disk. Alternatively, supply source code and have the publisher compile it—again, charge peanuts for the source code and big bucks for the compile rights. Another method is to add copy protection to the software.
2. Insist that the publisher commission your software.
3. Buy a modem. The SBE acknowledges that sales tax cannot be applied unless an exchange of a tangible medium takes place, and data transmitted by a modem are not tangible media. Why would the SBE propose such an illogical tax and at the same time provide an easy way to circumvent it? Perhaps a strong modem lobby exists in California.
4. Conduct all of your business transactions outside of California—or move to another state.

Max Stul Oppenheimer, PC, is a partner in the law firm of Venable, Baetjer & Howard, located in Baltimore, Maryland.

NEED IT FAST?
WANT TO KNOW MORE?
DON'T HAVE TIME TO SHOP AROUND?

INTRODUCING...

If you're like most system professionals you're up-to-date about the products in the PC marketplace. You're aware of brand and model differences, are informed about connectivity and compatibility problems, and you shop for competitive prices and fast service.

You're also probably among the many PC TECH JOURNAL readers who purchase by mail. That's why we're starting THE MART—PC TECH JOURNAL'S First Class Mail-Order Section.

Starting this month, and every month hereafter, you'll find the products you're looking for advertised in THE MART—and you'll benefit from the fast service and helpful support that identifies PC TECH JOURNAL advertisers.

If you're ready for First Class service, you're ready for THE MART.

WINTECH DATA PRODUCTS CORPORATION

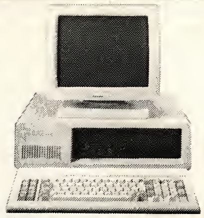
HIGH QUALITY, AMERICAN ENGINEERED & AMERICAN BUILT IBM COMPATIBLES UNBEATABLE PRICES!!

proteus-286 AT

100% FASTER 100% COMPATIBLE

TURNKEY SYSTEM
OUR PRICE $1745

SYSTEM INCLUDES:
- INTEL 80286 CPU, 6/8/10 MHz (12.5 MHz OPT)
- 1 WAIT STATE (0 WAIT STATE MHz OPT)
- 1024k RAM (MAINBOARD EXPANDABLE TO 4MB!)
- 5 MHz DMA BUS
- 3 SERIAL PORTS – RS 232 & RS 422
- 2 PARALLEL PORTS
- 12MB FLOPPY DRIVE
- HARD DISK/FLOPPY CONTROLLER
- 200 WATTS 110V/220V P/S
- CHOICE OF CGA COLOR GRAPHICS OR
- HERCULES COMPATIBLE GRAPHICS ADAPTR
- ON BOARD BATTERY BACKUP, CLOCK/CALENDAR
- AT KEYBOARD
- FULLY ASSEMBLED, ONE YEAR WARRANTY COMPLETE OPERATIONS MANUALS. FCC APPROVED!



proteus-turbo-xt

70% FASTER

TURNKEY SYSTEM OUR PRICE $895

SYSTEM INCLUDES:
- INTEL 8088-2 CPU, 4.77/8MHz
- 640K RAM (MAIN BOARD EXPANDABLE TO 1024K)
- 2 SERIAL PORTS
- 1 PARALLEL PORT
- SASI INTERFACE
- 360K TEAC FLOPPY DRIVE
- FLOPPY CONTROLLER ON BOARD
- CLOCK/CAL WITH BATTERY ON BOARD
- 135W P/S 110V/220V
- MONOCHROME GRAPHICS ADAPTER
- OR CGA COLOR GRAPHICS ADAPTR
- HIGH RESOLUTION MONOCHROME
- MONITOR (OPTIONAL COLOR MONITOR)
- FULLY ASSEMBLED. ONE YEAR WARRANTY. COMPLETE MANUALS. FCC APPROVED

DEALERS WELCOME.
CORPORATE & EXPORT ACCOUNTS WELCOME.

PHONE 201-288-8629
TELEX: 510 601 0960

HARD DISKS
SEAGATE

20MB	INTERNAL HALF HEIGHT		$350
30MB	INTERNAL	30MS	$670
40MB	INTERNAL	30MS	$CALL
60MB	INTERNAL	30MS	$CALL
120MB	MAXTOR	18MS	$CALL
260MB	CORE	18MS	$CALL

WINTECH DATA PRODUCTS CORPORATION

377 ROUTE 17, SUITE 618
HASBROUCK HEIGHTS,
NEW JERSEY 07604

PHONE:201-288-8629

TERMS: COD, VISA/MC, PREPAYMENT, L/C, OR WIRE TRANSFER.
(PRICES & TERMS SUBJECT TO CHANGE WITHOUT NOTICE)

RUNS ALL MAJOR IBM SOFTWARE INCLUDING AUTOCAD, FLIGHT SIMULATOR

MADE IN USA IMMEDIATE DELIVERY!!

Be 3-6 times more Productive!!! Guaranteed* For $99.95

Demo/Prototype/Application Generator

The "Assistant" helps you develop your functional specs (Generates Screen, File, Isam, Variable and Menu Control Documentation) and then allows you to "Quickly" prototype a validation model of your system. (Which can be incorporated as part of your functional specifications.) The Assistant can then generate a super-fast Turbo Pascal Program that features advanced screen input and control, a professional control menu, the database functions of (1) Add/Edit/Delete Records (2) Search Database by any Key (3) Database Recovery programs (4) Screen/Printer Report for each of the keys. Each Key can have up to 6 fields.

Our Users Report
- *"Since Fall of 85, I have generated over 300 program modules with it and find it to be just what I needed. Most all of the modules represent 5000 to 8000 lines of Pascal Code"* Oner Systems.
- *"By being able to produce a 21 screen and menu control demo so quickly helped me obtain the contract."*
- *"Speeded up my screen development by 6 times"* Elexor Associates.
- *"Has many of the features of the Super Mini development tools costing $10,000."* Applied Micro Systems.
- *"Saved months from having to recode portions of our system."* Real Green Inc.
- *"We developed 3 Vertical Market Applications in the 6 months we had your system."* Absolute Systems.

Btrieve Interface Module
Allows full record locking and Automatic file recovery for the industry's most popular LANs. Works with the industry's leader of professional databases for mulit-user LANs.
Requires Btrieve by SoftCraft Inc.

***RISK FREE TRIAL** Try the demo package included for 30 days. If not pleased return for a full refund.

Receive 6 Floppy Disks and a 200+ page manual containing:
1. **Screen Painter/Editor & Generator** • Paint menu screens using keyboard • Has variable dictionary to provide consistant edits • Date entry masks • Date & range checks • Field and/or global help screens • Box & line drawing • Error & message handler
2. **Help Screen Maker** - Different help screen for each field.
3. **Menu Editor & Generator** • Allows selection by 4 methods.
4. **Database Program Generator** • Produces "Easy to Read" code that can be easily modified by experienced developers.
5. **Resident Isam Module** - compatible with Turbo Toolbox, but saves 8K of codespace and 10K of dataspace.
6. **Resident Screen Capture Utility** which allows you to capture Text Screens from any running program.

& Much, Much More

NO ROYALTIES on Generated Programs

Credit Card & C.O.D. Orders Call: 1-800-821-9503
In Florida 1-800-342-0137

☐ Turbo Master Assistant by Hawaiian Village Software	$99.95
☐ Btrieve Interface by Innovative Interfaces	99.95
☐ Turbo Pascal by Borland International	$69.95

For Other Inquiries Call:
(305) 892-5686

Add 7.50 shipping to all U.S. Cities. All foreign orders add 15.00 per product ordered

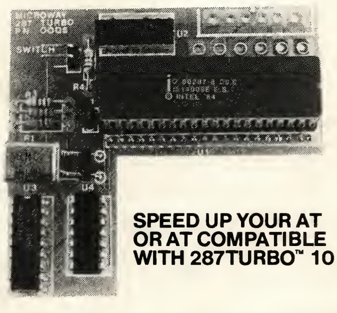
A MEGABYTE FOR DOS!

MicroWay is the world's leading retailer of 8087s and high performance PC upgrades. We stock a complete selection of 8087s that run from 5 to 12 MHz. All of our coprocessors are shipped with a diagnostic disk and the best warranty in the business – one year! We also offer daughterboards for socketless computers (NEC PC) and 287Turbo which increases the clock speed of the 80287 from 4 to 10 MHz. Our NUMBER SMASHER/ECM™ runs at 12 MHz with a megabyte of RAM and achieves a throughput of .1 megaflops with 87BASIC/INLINE, Intel Fortran, or Microsoft Fortran. Software reviewers consistently cite MicroWay software and 8087 expertise as the best in the industry! Our customers frequently write to thank us for recommending the correct software and hardware to meet their specific needs. They also thank us for our same-day shipping! In addition to our own products which support the 8087 and 80287, we stock the largest supply of specialized software available. For more information call us at

617-746-7341

NUMBER SMASHER/ECM™ THE FASTEST ACCELERATOR CARD AVAILABLE

gives you 12 MHz speed in two modes: 704K or one megabyte of "Extended Conventional Memory." MEGASWITCH MMU and MegaDOS software make it possible to run DOS applications with up to 1015K using PC compilers, AutoCAD and Lotus 1-2-3. Does not require EMS software. Totally compatible. Priced from **$599** with 512K to **$1199** for complete package. Optional 8087-12 . . . **$295**



SPEED UP YOUR AT OR AT COMPATIBLE WITH 287TURBO™ 10 MHz

Micro Way® 8087 Support

For the IBM PC, PC XT, PC AT and Compatibles.

A2D-160™ MicroWay's Data Acquisition Board performs 160,000 12 bit Analog to Digital conversions per second! Includes software drivers. The fastest 12 bit A to D board available. For the IBM PC XT and compatibles. **$1295**

87SFL™ MicroWay's Scientific Function Library contains 170 scientific and engineering functions. Callable from most 8087 compatible compilers . . . First Language **$250**; Additional **$100**.

MATRIXPAK™ manages a **MEGABYTE!** Written in assembly language, our runtime package accurately manipulates large matrices at very fast speeds. Includes matrix inversion and the solution of simultaneous linear equations. Callable from RM or MS Fortran, MS Assembler, or 87BASIC/INLINE. each **$99**

87FFT™ Written in assembly language, performs Forward and Inverse FFTs on real and complex arrays which occupy up to 512 Kbytes of RAM. Also does convolutions, auto correlations, hamming, complex vector multiplication, and complex to radial conversions. Callable from most 8087 compatible compilers **$200**

87FFT-2™ performs two-dimensional FFTs. Ideal for image processing. Requires 87FFT **$100**

FASTBREAK™ employs the 8087 to increase the speed of Lotus 1-2-3™ Version 1A or 1A* by up to 36:1 . . . . **$79**

87 Verify™ For users who have to be absolutely sure of their results! This background task periodically performs an 8087 accuracy and stress test. **$49**

Microsoft Fortran V 3.31	**$209**
IBM Professional Fortran	**$565**
Ryan-McFarland Fortran V 2.0	**$399**
NAG Fortran Library	**$300**
Grafmatic for Fortran or Pascal	**$125**
MultiHalo Graphics (1 language)	**$189**
LABTECH NOTEBOOK	**$745**
UnkelScope	**$549**
INTEL ABOVE BOARD	**CALL**
JRAM, AST, MAYNARD	**CALL**

MegaPage™ The only Intel-Lotus EMS board which comes with two megabytes of cool-running, low power drain CMOS RAM installed. Includes RAM disk, print spooler, disk cache, and EMS drivers. For the IBM PC, XT & compatibles **$549**

MegaPage AT/ECC™ EMS card for the PC AT and compatibles includes Error Correction Circuitry. With ECC, 11 RAM chips cover 256K so the user never encounters RAM errors. With 1 megabyte CMOS **$799**; with 3 megabytes CMOS **$1295.** Optional serial/parallel daughterboard. . . . . **$95.**

DFixer™ Our disk utility which thoroughly checks PC or AT hard disks for bad sectors and updates the MS DOS file allocation table accordingly. Solves the AT hard disk problem! . . . **$149**

DOptimizer™ Optimizes the way your hard disk or floppy stores its files. Speeds up accesses by recombining fragmented files. **$49**

DCache™ Our disk caching software speeds up your I/O by storing repetitively used tracks in memory. The amount of memory used can be selected in 64 Kbyte banks **$49**

87MACRO/DEBUG™ Contains all the pieces needed for writing 8087/80287 assembly code & MicroWay's 87DEBUG debugger. **$199**

OBJ→ASM™ A multipass object module translator and disassembler. Produces assembly language listings which include public symbols, external symbols and labels commented with cross references. Ideal for patching object modules for which source is not available. **$200**

87BASIC™ includes patches to the IBM BASIC or MS Quick BASIC Compiler for USER TRANSPARENT 8087 support. Provides super fast performance for all numeric operations including trigonometrics, transcendentals, addition, subtraction, multiplication, and division. . . . . . each **$150**

87BASIC/INLINE™ converts the output of the IBM BASIC Compiler into optimized 8087 inline code which executes up to seven times faster than 87BASIC. Supports separately compiled inline subroutines which are located in their own segments and can contain up to 64 Kbytes of code. This allows programs greater than 128K! Requires the IBM BASIC Compiler Version 1 and a Macro Assembler. Includes 87BASIC . . . **$200**

MICROWAY UDI runs RTOS or RMX compilers under DOS **$300**

8087 UPGRADES

All MicroWay 8087s include a one year warranty, complete MicroWay Test Program and accurate installation instructions.

8087 5 MHz **$109**
For the IBM PC, XT and compatibles.

8087-2 8 MHz **$149**
For Wang, AT&T, DeskPro, NEC, Leading Edge.

80287-3 5 MHz **$179**
For the IBM PC AT and 286 compatibles.

80287-6 6 MHz **$229**
For 8 MHz AT compatibles.

80287-8 8 MHz **$295**
For the 8 MHz 80286 accelerator cards.

NEC V20, V30	**$16, $30**
64K RAM Set 150ns	**$10**
256K RAM Set 150ns	**$29**
256K RAM Set 120ns	**$39**
128K RAM Set PC AT	**$49**

287Turbo™ 10 MHz If you own an AT, Deskpro 286 or AT compatible, this is the card you need to get reasonable numeric performance. It plugs into your 80287 socket and includes a specially driven 10 MHz 80287. The card comes in three configurations. The IBM AT version includes a hardware RESET button **$450**

287Turbo 8 MHz **$369**

87/88Turbo™ is a stubby card which includes a clock calendar and a speed controller which changes the speed of your motherboard from 4.77 to 7.4 MHz. Its use requires your PC to have a socketed 8284. Typical speed increase is 1.6 to 2.0. The card overcomes slow hardware by slowing up only when such devices are accessed and running at full speed otherwise **$149**
Optional 8087-2 . . . . **$149**

286TurboCache™ This new MicroWay accelerator uses 8K of cache memory and 80286/80287 processors to provide an average speed increase of 3:1 for most programs. Call for specifications and benchmarks **$595**

Call for our complete catalog of software which supports the 8087.

In London, please phone 223-7762

Micro Way

P.O. Box 79
Kingston, Mass.
02364 USA
(617) 746-7341

PC BRAND™: CAREFULLY CHOSEN PROGRAMMER TOOLS

30-DAY MONEY-BACK GUARANTEE
We refund the purchase price of any product returned within 30 days in entirely resalable condition. **You can even try out programs themselves** *if product code begins with E, T, or L through N — even if it means breaking the disk seal. Some developers do pose limits, so for products beginning with other letters, opening sealed disks constitutes acceptance. But you can at least review the manual. There's just nothing stopping your buying from PC Brand.*

BRIEF *Is Anything But. A Whopper of an Editor*

With a name that belies its thoroughness, Brief™ has every feature you've ever contemplated for your editor-in-chief. Text, from keyboard or files, is housed in multiple buffers, and scrolled through one or more windows you open, close, resize. A text buffer may be called to different windows to view two areas at once. A change in one changes both. Text blocks may be marked for printing, writing to files, movement to scrap buffers for cut and paste into other buffers, or deletion, with as many "undo" levels as you want.

Brief has text search abilities rivaling "grep", with wildcards for matching, indifference to intervening characters, acceptance of character ranges.

If you use Lattice, C86™, or Wizard, and have 320k, you can compile your C program without ever leaving Brief. It finds the lines with errors, and marches you through the text for repairs.

Parts of Brief were written with its own Lisp-like macro language which has structure, 32-character variable names, conditional execution, loops, and you can actually read it! Nothing like the hieroglyphs we've seen elsewhere. Bulletin board and public domain disks with macros. "Simply the best text editor you can buy", Dvorak *Infoworld*. (Needs 192k.)

Ask for:	List:	PC Brand:
U0590	$195	**Call**

HALO GRAPHICS SYSTEM *Multi-Board Graphics Library*

The premier graphics library that got the ball rolling for PC-based graphics and has grown so omnipotent that it supports over 25 graphics boards — including IBM's EGA and Nr. 9 Revolution's hi-res series — and has a multitude of mouse and printer drivers. All that in each box. Separate C versions for Lattice, M'soft, Aztec, C186. What does Multi-Halo do? A down to the last pixel graphics library plus functions to reset drivers so distributed program can run on anything. Wonderful value for single license. Costly royalties though for redistribution. Specify: S0315 & Language. List: **$300.** We: **$219.** With Dr. Halo II, a free-standing "paint": List: **$440,** Us: **$299.**

WINDOWS for C/WINDOWS for DATA *Microsoft Windows™ and TopView™ Compatible*

CALL FOR FREE DEMO!

Windows for C™ is a library of over 80 functions to add the pizzazz and practicality of window partitioning to your application. Unlimited windows, each defined in a C structure for easy reference throughout your program, can be made either to pop up or permanently overwrite the screen. Routines will scroll and highlight lists with arrow keys, will read and scroll ASCII files vertically and horizontally in windows, and even write to memory-loaded files off the screen.

Logical treatment of video attributes permits unchanged programs to run on color or monochrome. Colors of windows are set individually.

All functions are in separate modules; only those used are linked. Only buffers holding on-screen or temporarily obscured windows occupy RAM; others released dynamically. Best overall rating and fastest display in Bill Hunt's 7/85 *Tech Journal* review of five windowing products.

Windows for Data comprises all of Windows for C but takes in data through the windows as well. At the high level a single function lets you specify prompt string, field length, data type, screen location, picture, target variable, then sets lesser functions scurrying to get and process a user's input. There are utilities to get system date and time, mess with strings, create your own masks for fields.

Field options can require entry, prevent entry, permit insert or overtype, beeping on invalid or overflow keystrokes, and attachment of field-specific help messages and functions you want called to display messages or validate entries. And you decide which keys will clear a field, jump to the next or prior, quit, etc. Options diverse enough that a set of "fields" can be made to behave like a Lotus™ menu.

Specify Compiler:	List:	PC Brand:
T0100 Windows for C	$195	**$149**
T0150 Windows for Data	$295	**$250**

C-WORTHY LIBRARY *Fits Out Applications with Shipshape Interface*

FREE DEMO!

Not the usual flotilla of functions for small crafting, but a formidable battle wagon for major C engagements.

C-Worthy™ wraps an entire user interface around your application. Its full power can be summoned by only a few high level calls. Proof? A single function call sets up a complete text editor in a screen window!

- High level calls pop menus and scrollable lists to the screen, restoring the background when dismissed, and branching to the chosen activity in your application.
- Windowing facilities open portholes of up to screen size for viewing virtual screens larger than the physical screen.
- Full on-screen context-sensitive help management. Keyboard entry routines look for the help key and interrupt with paged text windows explaining what to do.
- Full error message interface sends error codes to C-Worthy which discusses problem with user. Gone is all that error-checking clutter from your core program.

Your application is nested in this powerful environment. C-Worthy's architecture uses C's pointers to functions to drive your application.

C-Worthy includes utilities to maintain text files of help and error messages. Their segregation means applications can readily be translated into foreign languages without program rewriting. Plus display routines resize for text length.

Decomposed low level functions used by the high-level calls are supplied. All machine dependency is housed in interchangable overlays loaded at run-time. Thus no recompilation needed to run on a mix of PC and MS-DOS machines.

C-Worthy hands you a consistent and intuitive interface and a revolutionary design approach. Novell found it "played a key role and accelerated development" making NetWare™ easier to use. "You owe it to yourselves to take a look."

Specify Compiler:	List:	PC Brand:
T0500	$295	**$269**
T0550 Network License	$495	**$449**

C-TREE *B-Tree File Manager, Source Code, No Royalties!*

C-tree is sturdy code that has weathered many seasons of prolonged and widespread use. It comes in C source, so you can modify it to fit a special case. No royalties provided you bind it into your binary application.

C-tree's design splits nodes to allow any number of users to access an index file simultaneously even when updates are in progress. So multi-user configurations and adaptation to networks are possible. Record-locking routines are provided for DOS 3.1/3.2, UNIX and XENIX.

Thanks to source code which does not deviate from the K&R standard, C-tree can travel. Tests in many environments prove that C-tree gives your application a ticket to anywhere.

C-tree permits any number of keys for a data file, supports duplicate keys, alphanumeric or numeric, supports files of variable record length; multiple keys in one index file, and keys of variable length. Both high level ISAM routines which handle details with minimum coding, and decomposed step-by-step functions you can access directly. It's comprehensive.

Ask for:	List:	PC Brand:
F0660	$395	**$329**

CURSES *Unix Style Screen Management*

Curses from Lattice™ manages the screen of the PC like Unix™ curses. Library of 84 functions and macros parallels Unix with matching parameter lists. So Unix programs are at home on the PC, and vice versa. Keeps any number of screens in memory, supports color, vast function set to get characters, wrap lines, scroll, blank lines, highlight, etc. Like Unix refreshes screen only on your command. Ask for: L0850. List: **$125.** Here: **$99.** With Source: L0860, **$250/$199**

dBC *Lattice Library Maintains dBASE Compatible Files With the Power and Speed of C*

dBC™ links C to dBASE. It creates and maintains files and their indexes which exactly replicate dBASE file design. So dBASE can read and update them. And the reverse. dBC can use any files created by dBASE. Now C and dBASE can operate on the same data bases interchangeably.

That opens up the widespread culture of dBASE installations to exploitation by C programmers. Tap that market, avoid the resident dBASE language, and gain the advantages of C with this single product.

dBC's functions parallel all dBASE's file handling commands, many decomposed to give closer control. Each backed by demo source files on disk.

Use dBC for custom work for clients, or on its own. It's a complete ISAM file manager for C whether or not dBASE will be used in tandem, supports all four memory models, and can have sixteen index and data files open. Big discount to buyers of both dBASE II and III versions. Specify Lattice, Microsoft 3.x, or DeSmet.

Versions:		List:	PC Brand:
L00II	For dBASE II	$250	**$195**
LCCII	With Source	$500	**$390**
L0III	For dBASE III	$250	**$195**
LCIII	With Source	$500	**$390**

PANEL *Feature-Laden Screen Design Tool*

Writing your own screenware can blow completion dates and profits. Panel™ works with you interactively to set up foolproof screen displays and data entry forms rapidly. Output is C source code.

Not just single plane: layer your screen designs with up to ten overlapping images: Background pop-up lists, help boxes, and alternate input fields.

Panel builds in a user interface for keystroke movement within and between fields, supplies validation routines for checking user field entries. Diverse attributes may be selected for any field — size, data type, color, conversion of input to upper case; clearance of existing data when new entry is started; masks for standard formats (eg, dates); phrases which fill in when their first letter is typed; multiple-choice lists from which to choose by cursoring a highlighted bar. Fields may be multi-lined and scrolled if larger than the screen space allotted them. Specify: S0400 & Compiler. List: **$295,** Us: **$229**

WHY US?: Latest versions of all products • Everything listed is in stock • Shipped 24 hours or sooner • No surcharge for credit card or COD purchase. **NEED TERMS?** On-the-spot credit to most public companies, government, educational, medical institutions. **LOOKING FOR SOMETHING?** We can get many more products — just ask! **NEED MORE INFO?** Our Catalog and literature cover just about everything.

For Orders, Literature, or Catalogs, Call Us at...

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

1986 PC BRAND
Prices, terms, and specifications subject to change without notice.

SAVE $5 Just tell us this code **J8**

TODAY's TOP QUALITY AIDS TO PROGRAMMING PRODUCTIVITY

GREENLEAF FUNCTIONS *Bountiful Harvest*

C source, assembler source, and binary libraries of 225 functions for many compilers. Emphasizes tight functional groupings to minimize loading code which your application may never use. Manual helps select functions, bulletin board, too.

A sampling: *DOS* extensions for file and directory manipulation; *Screen:* to select mode, page, monochrome or color, palette; cursor shape, positioning; clearing and scrolling; pixel get and put; read light pen. *Strings:* Center, justify, etc.; efficient list operations which add, delete, sort string pointers for top speed. *Other:* graphics character primitives, keyboard status, function key assignment, time/date, read registers and memory size, peek and poke. Mature best-seller. Specify: S0770 & Compiler. List: **$185**, Here: **$139**

PFORCE *Phoenix Pfunction Pfestival*

Lotus® didn't do badly pulling it all together in one place. Phoenix has followed suit with the ultimate integrated C library, offering everything from low level functions for hardware access to complete b-tree database management. Along the way are prerequisites such as string manipulation, time/date, field and screen editing, but also four styles of menus (Lotus included), windowing, background tasking, DOS interfaces, directory management, even interrupt-driven communications. Design emphasizes objects, so characteristics of windows, databases, records and fields can be initiated and changed outside functions.

One large collection in place of bits and pieces means one set of instructions and PforCe™ has tutorials, extensive examples, quick reference, and on-line help.

Everything in source, no royalties, all memory models of Lattice, M'soft. Specify: S0220 & Compiler. List: **$475**, PCB: **$349**

GREENLEAF COMMUNICATIONS *Hello World*

Want your application to communicate with other users or remote date bases by asynchronous communications built right into your C programs! Even if you don't need it now, that's a skill to have at the ready!

120 functions *and demo programs in both C and assembler source code* set up separate transmit and receive ring buffers for up to 16 simultaneous channels. Interrupt driven so you can halt an incoming record, display it, file it, let the user edit it, then continue. Goodbye separate communications software.

Supports up to 9600 baud, ASCII or binary, any parity or word length, 8250 UARTs, Xon/Xoff and Xmodem, WideTrack receive. Specify: S0750 & Compiler. List:**$185**, Us: **$139**

PRE-C *Pick the Lint from Your Program*

Pre-C is like UNIXs lint. It finds problems your compiler won't. Problems that a debugger will have trouble figuring out. Even problems which will cause trouble with other compilers.

Compilers see one module at a time. Modules only meet at link time. Pre-C looks at *all modules at once* and reports conflicts in data type declarations; function call parameters which disagree with functions, machine-dependent expressions which inhibit portability. It spots obsolete usage (even C changes), casts with suspect conversions, variables never used, functions never called, unreachable code. Adheres to UNIX System III compile standard to ensure your portability. Ask for: P0590, List: **$395**, Ours: **$279**

DAN BRICKLIN'S DEMO PROGRAM
Storyboard Your Program

The Legendary One has created Metaphor Two when the rest of us are still on Zero. Dan's first was the original electronic spreadsheet (VisiCalc™). This one is for programmers.

Words don't express program ideas because programs are screens! Dan's Demo creates slide shows. Create a screen — a snapshot of your planned product as it runs. Anything goes: words, borders, box rules, inverse and underlining of monochrome, fore- and background color. Copy this "slide" to an empty screen. Change it a little, to show the next instant of run-time. Do it again. Presto, a whole slide show of your program in action.

All 250 characters and attributes are available from scrollable lists which pop to the screen. All commands are layered in Lotus-style pop-up menus. Frequent choices mapped to function keys as well. 80x25 character mode, not bit-mapped.

Screen areas can be blocked for cut and paste or filled with color or characters, even blink. Slides can overlay on others, can be shuffled, deleted. Slides can proceed at time intervals or branch anywhere in the slide sequence depending on user keyhits.

Invaluable to prototype the program you are about to write, to position the labels, choose the color decor, smoothe out the keystroke interface. Or load the "capture" utility and snapshot the screens of any running program for an instant slide show.

Each copy entitles you to redistribute fifty of the slide projector program that runs demos. Plain manual, no binder keeps price of big product small. "Might... become *the* essential tool in... user interface prototyping," *Tech Journal*. Ask for: N0100. List: **$75** US: **$69**

BASTOC OPTIMIZES!
Translates BASIC Into C

For a trifling price, BASTOC™ moves truckloads of BASIC code over to C. It's a translator which takes in Microsoft Extended BASIC and emits pure K&R C for Lattice 3.0. It will optionally convert your program into a single monolithic C function or decompose it into separate functions, one for each GOSUB label.

Version 2's optimization dramatically reduces execution time. Converts to integers those variables in BASIC programs which do not need floating point. Where BASIC uses full assignment statements to increment counters, BASTOC converts to C's compact form. Strings dynamically allocated ridding your application of BASIC's catatonic halts for garbage collection. Creates structure of even convoluted BASIC code. Huge worksaver.

Ask for:	List:	PC Brand:
S0375	**$495**	**$399**

Shopping List for the Power Workbench

	LIST	OURS
ASSEMBLERS & DEBUGGERS		
Advanced Trace-86 *Morgan, ASM Interpreter*	175	**149**
Codesmith-86 *Debugger by Visual Age*	145	**109**
Cdebugger *by Micro-Software Developers*	165	**139**
CSD Debugger *C source level by Mark Williams*	75	**75**
C-Sprite Debugger *by Lattice, source level*	175	**139**
Microsoft Macro Assembler *with Utilities*	150	**109**
Periscope I Debugger *Data Base Decisions*	295	**269**
Periscope II *Data Base Decisions*	145	**129**
Pfix86 *by Phoenix, Assembly level debugger*	195	**149**
Pfix86 Plus *by Phoenix, Symbolic Debugger*	395	**279**
The Profiler *DWB Associates, with Source*	125	**99**
BASIC LANGUAGE		
BetterBASIC *Summit Software*	195	**165**
BetterBASIC Utilities *8087 Math Support*	99	**85**
Btrieve Interface	99	**85**
Run-Time Module	250	**225**
Microsoft BASIC Interpreter *for XENIX*	350	**295**
Microsoft QuickBASIC Compiler *full BASICA*	99	**79**
Professional BASIC *by Morgan*	99	**79**
8087 Math Support	50	**47**
True BASIC *True BASIC Inc*	150	**119**
Run Time Module	500	**420**
True BASIC Libraries *Btrieve, Asyn, Sort, etc.*	Var	**Call**
C COMPILERS		
C-86 Compiler *Computer Innovations*	395	**289**
Lattice C Compiler *from Lattice*	500	**299**
Let's C Compiler *by Mark Williams*	75	**69**
with CSD Source Level Debugger	150	**129**
MWC-86: Mark Williams C Development	495	**369**
Microsoft C Compiler	395	**259**
C INTERPRETERS		
C-Terp *by Gimpel Software*	300	**249**
Instant C *by Rational Systems*	500	**395**
Interactive-C *by IMPACC with debugging*	249	**219**
RUN/C Professional *from Lifeboat*	250	**185**
RUN/C *without Loadable Libraries*	120	**109**
TEXT EDITORS		
Brief *from Solution Systems*	195	**CALL**
CVUE *by Lattice*	75	**69**
with Source	250	**195**
Epsilon *by Lugaru Software, like EMACS*	195	**169**
Fast C *by Lifeboat*	130	**105**
FirsTime *by Spruce Technology, C syntax*	295	**229**
Kedit *by Mansfield, similar to Xedit*	125	**115**
LSE, the Lattice Screen Editor *Multi Window*	125	**100**
Pmate *by Phoenix, with Macros*	225	**159**
Text Management Utilities *Grep, splat, diff, etc.*	120	**100**
Vedit *by Compuview*	150	**119**
Vedit Plus *by Compuview*	225	**180**
FILE MANAGERS		
Btrieve *by Softcraft, no royalties*	250	**195**
Btrieve Network *by Softcraft*	595	**465**
c-tree *by FairCom — no royalties, source*	395	**329**
dBC *dBASE file manager from Lattice*	250	**195**
with source	500	**390**
dbVista *single user DBMS by Raima*	195	**159**
with source	495	**429**
dbVista *multi-user DBMS*	495	**429**
with source	990	**849**
Opt-Tech Sort *Can sort Btrieve files*	149	**119**
SCREEN DESIGN		
Curses *by Lattice, UNIX screen designer*	125	**99**
with Source	250	**199**
On-Line Help *from Opt-Tech Data*	149	**119**
Panel *by Roundhill, no royalties*	295	**229**
View Manager for C *by Blaise*	275	**209**
Windows for C *Vermont Creative Software*	195	**149**
Windows for Data *includes Windows for C*	295	**250**
ZView *Data Management Consultants*	245	**199**
GRAPHICS		
Essential Graphics *by Essential, no royalties*	250	**210**
GSS Graphics Development Toolkit	495	**399**
GSS Kernel System *by Graphic Software*	495	**399**
GSS Kernel System *for IBM RT*	795	**676**
GSS Metafile Interpreter	295	**249**
GSS Plotting System	495	**399**
Halo Graphics Kernel System	300	**219**
with Dr. Halo II, by Media Cybernetics	440	**299**
COMMUNICATIONS		
Asynch Manager *by Blaise, for C or Pascal*	175	**149**
Greenleaf Communications *by Greenleaf*	185	**139**
Software Horizons Pack 3	149	**119**
UTILITY LIBRARIES		
Blaise C Tools 1	125	**109**
Blaise C Tools 2 *Call about combined price*	100	**89**
C Food Smorgasbord *by Lattice*	150	**109**
C Utility Library *by Essential, 300 functions*	185	**139**
Greenleaf Functions *by Greenleaf Software*	185	**139**
PforCe *by Phoenix, vast library*	475	**349**
Software Horizons *Packages*	Var	**Call**
TopView Tool Basket *by Lattice, source avail*	250	**199**
Vitamin C *by Creative Programming*	150	**139**
DEVELOPMENT TOOLS		
Code Sifter *by David Smith Software, Profiler*	119	**99**
C-Worthy *by Custom Design Software*	295	**269**
C-Worthy for Network *Menus, help, errors*	495	**449**
Dan Bricklin's Demo Program *Prototyper*	75	**69**
LMK from Lattice *by Lattice, "make" like UNIX*	195	**149**
PC-Lint *by Gimpel Software, after UNIX's "lint"*	139	**125**
PFinish *by Phoenix, EXE performance analyzer*	395	**279**
Plink86 *by Phoenix, Overlay Linker*	395	**279**
Plink86 Plus *Utilizes memory for overlays*	495	**359**
Pmaker *by Phoenix, like UNIX "make"*	195	**149**
Pre-C *by Phoenix, UNIX "lint"-alike*	395	**279**
Pfantasy Pac *six Phoenix products*	1295	**895**
OTHER TOOLS		
BASTOC *by JMI, convert BASIC to C*	495	**399**
BASIC-C *BASIC's functions added to C*	175	**139**
The HAMMER *by OES Systems*	195	**179**
PASM86 *by Phoenix, Macro Assembler*	295	**219**
PTel *by Phoenix, Binary File Communicator*	195	**149**
Rtrieve *by Softcraft, Btrieve Report Generator*	85	**75**
Xtrieve *by Softcraft, Query Utility for Btrieve*	195	**175**
FORTRAN COMPILERS & UTILITIES		
ACS Time Series *by Alpha Computer Service*	495	**469**
Forlib- Plus *by Alpha Computer Service*	70	**59**
Microsoft FORTRAN *Links with Microsoft C*	350	**219**
Microsoft FORTRAN *for XENIX*	495	**389**
Pro FORTRAN *by Prospero*	390	**345**
RM/FORTRAN *by Ryan-McFarland*	595	**399**
Scientific Subroutine Library *by Peerless*	175	**149**
Scientific Subroutine Package *by Alpha*	295	**269**
The Statistician *by Alpha Computer*	295	**269**
Strings & Things *by Alpha Computer*	70	**59**
OTHER LANGUAGES & UTILITIES		
Microsoft COBOL Compiler	700	**499**
Microsoft COBOL Compiler *for XENIX*	995	**795**
Microsoft COBOL Tools *with Source Debugger*	350	**259**
Microsoft COBOL Tools *for XENIX*	450	**333**
Microsoft Lisp *New Common Lisp*	250	**189**
Microsoft MuMath *Includes MuSimp*	300	**199**
Microsoft Pascal Compiler *Links with M'soft C*	300	**199**
Microsoft Pascal Compiler *for XENIX*	495	**415**
Pro Pascal *by Prospero, ISO Validated*	390	**345**

PRICED TO SAVE YOU MONEY, SHIPPED FAST ANYWHERE.

BEST PRICES YET!

INTERACTIVE–C

NEW!

Compiler-Compatible Interpreter, Editor, Debugger

Earlier C interpreters were miraculous compromises. Interactive-C™ shows how far C interpreters have come. More than an interpreter, Interactive-C is a fully-integrated development environment: a complete K&R interpreter bound tightly to its own editor and debugger.

Slice through programming projects like a hot knife through butter. Extensive error-checking insures immediate detection of program misbehavior. State of the art debugging tools include breakpoints, watchvalues, several stepping options and interactive viewing and modification of variables. An Interactive-C exclusive lets you interrupt to edit and "continue" from where you left off. Eliminates plodding replays of already debugged code — the ball and chain of other interpreters.

Operate Interactive-C using adjustable edit, command, and status windows. Toggle a second screen showing only your program's output — never any crowded intermixing. Or, boost productivity with twin CRTs. Load object code of functions you have already compiled. Or of commercial libraries. Interactive-C has immediate mode, syntax checking both as you type and run, and cursor positioning precisely pointing at an error, not possible with incremental or pseudo-compilers which leave source code behind.

100% compiler compatible — right down to header files and library calls. Port programs between Interactive-C and your compiler with no modifications whatever — not even tricky areas of dynamic memory allocation and I/O.

Specify:	List:	PC Brand:
E950 & Compiler	**$249**	**$219**

RUN/C PROFESSIONAL

C Interpreter Links Binary Libraries

Run/C comes in an apprentice and pro version. The professional model dynamically loads and unloads multiple binary function libraries like C-Food Smorgasbord™ and Halo Graphics™ — potentially any library compiled with Lattice's large model. Inside this interpreter your C program can reach for functions in the best of commercial libraries.

This C interpreter behaves like PC BASIC meets WordStar®. Use fullscreen editing to create a program. RUN it. If it stumbles, LIST it, EDIT it, RUN it again, fix it again. Use familiar commands like LOAD, MERGE, SAVE, FILES, even TRON and TRACE.

Ideal for program development. Put up code at high speed, try out things devil-may care, let RUN/C find your malaprops. Blast away until tight little code segments are undyingly faithful.

Lots more features: system interrupts, a shell command to invoke any operating system command without leaving Run/C, debugging aids ingeniously installed as a Run/C function. Call for debugging conditionally from within your program, a specific function or a menu of aids including immediate mode, single-step tracing, changing of variable values.

Manual shows how to develop the interface to a commercial library, using the Lattice compiler (a must!). Link your own function archive the same way. (320k minimum; 512k recommended to fit libraries.)

Ask for: S0950 List: **$250** PCB: **$185**

PLINK86 & PLUS

Cached Overlays Maximize Memory Use

Long the overlord of overlay linkers, standard Plink86 shoehorns large programs into small machines by sharing memory, swapping program segments in from disk. A 512k program could run in a 128k machine, for example. The Plus version goes beyond: if it finds itself in a larger machine, it moves whatever program overlays that fit into leftover memory. Overlays now swap at memory speed not disk speed. It also can automatically restore a displaced overlay to which a subsequently called overlay must return, and assign library modules to a program's root segment or its overlay areas.

Plink86-Plus: List: **$495.** Us: **$359**. Plink86: List: **$395**. Us: **$279**

LATTICE C COMPILER

Major Upgrades to the Best Selling C Compiler

Lattice now embraces key UNIX™ enhancements which have entered the language since K&R: void functions returning no value, enumerated data types to assign stepped values to variables, data passing between structures by assignment.

The greatly expanded libraries (325 functions!) enable the file sharing and record locking provisions of DOS 3.1, provide a full complement of transcendentals, and a host of utilities to mimic the UNIX and XENIX™ environments.

Lattice 3.0 defaults to the ANSI proposed standard when you need strict adherence, but command line options restore leniency. And it adopts ANSI checking of external function arguments by data type to kill bug swarms when modules join up at link time.

Lattice now delivers smaller .EXE files, boasts very fast link times and a more efficient aliasing algorithm. New options generate code to use 80186 and 80286 features; 8087 of course sensed and utilized. Lattice has enjoyed pre-eminence so long that developers have created far more snap-on tools for Lattice C than any other compiler. William Hunt's *PC Tech Journal* review of 12 compilers awarded Lattice the only "very good" rating for add-on library availability.

Ask for:	List:	PC Brand:
S0100	**$500**	**$299**

FOREIGN POLICY

We ship anywhere. Phone or Telex your order. Credit cards: We need card number, expiry date, name and address of card. Or wire funds to PC BRAND, c/o Chemical Bank, 126 East 86th St., NY NY 10028, Account No: 034-016058. **We'll ship immediately** and confirm by Telex if you provide number.

BETTER BASIC

Convert Microsoft BASIC. Structured, Compilable.

Combines the familiarity of BASIC with the best features of C, Pascal, and Modula 2, yet BetterBASIC is 100% compatible with Microsoft's GW™ BASIC and IBM BASICA including graphics, sound, and assembly language calls. So load your old programs and RUN. SAVE and they are converted automatically to BetterBASIC!

It's big: Needs 192k; programs can go to the PC's full 640k. It's comfy: Behaves like M'soft BASIC at the interactive level, with a full-screen editor, direct statement execution, and always poised to RUN. It's fast: Each statement checked and compiled once, not every time encountered. Sieve runs *6 times faster* than with M'soft.

C-like structures house file records so goodbye to FIELD, MKI$, CVD, LSET, etc. Named "procedures" replace GOSUBs to linenumbers. Lots more features: built-in linker for compiled modules; trace; debugging breakpoints; cross-reference command; 32k strings; DOS and BIOS calls and interrupts; recursion. Run-time module stores object code for redistribution.

Ask for::		List:	Us:
S1200	BetterBASIC	**$195**	**$165**
S1201	Run-time Module	**$250**	**$225**
S1202	8087 Interface	**$ 99**	**$ 85**
S1205	Btrieve Interface	**$ 99**	**$ 85**

GSS GRAPHICS SYSTEM

Leave the Device Driving to GSS

ANSI CGI STANDARD!

GSS™ has reconfigured two components of its comprehensive graphics tools to conform with the ANSI Computer Graphics Interface (CGI) standard.

At the heart of the system is the Development Toolkit which contains all language interfaces and device drivers for keyboards, mice, joysticks, tablets, printers, plotters, cameras, and more. Drivers house management of vector graphics (plotters) and bitmaps used by raster input devices (scanners) to insulate the application program from concern for device idiosyncracy. No one else has implemented CGI that way. It means your programming remains generic; just switch drivers and the same program will drive a different device.

GSS Kernel™ conforms to level 2b of ANSI's Graphical Kernel System (GKS) and contains all its needed drivers and language bindings. Kernel has macro level tools to draw and color an object, store the sequential instructions, and recreate the object on its own, as well as segment it, transform it, etc. So powerful, a single command may represent several score lower level statements.

Plotting has the equivalent GKS tools for graph and chart generation and their captioning: hand it apples and oranges, say "pie", and it bakes the numbers into a digestible display for screen or plotters.

Kernel and Plotting have tools to convert images they create to ANSI Computer Graphics Metafiles (CGMs), a tokenized standard for storing every form of graphic image as data. The Metafile Interpreter reads the contents of a CGM and interprets it with full CGI capability for recreation on various devices.

Quality software? IBM thinks so. They sell the GSS series under their own label.

Unit royalties and annual fees have been instituted for redistribution. Needs 256k.

Ask for:		List:	PC Brand:
GS010	CGI Dvlpmt Toolkit	**$495**	**$399**
GS020	Kernel System	**$495**	**$399**
GS025	Kernel for IBM RT	**$795**	**$676**
GS030	Plotting System	**$495**	**$399**
GS040	Metafile Interpreter	**$295**	**$249**

BTRIEVE

ASK ABOUT XTRIEVE & RTRIEVE

Queen B-tree File Manager Abdicates Royalties

There's no longer a tithe to incorporate Btrieve™ in applications, a welcome proclamation if royalties would ruin your profit margins. Btrieve takes complete charge of all file creation, indexing, reading, writing, insertion, deletion, space recapture, forward and backward searching. It builds function call "commands" right into the language you use: interfaces to C, Pascal, BASIC, and COBOL, with sample programs in all four, come with each copy.

Btrieve has mainframe specifications! Its balanced-tree indexing scheme finds any key in a million in four or less accesses. Files may have up to 24 indexes; fixed record length to 4090 characters; indexes up to 255 characters; files of 4 billion bytes. Can even extend a file across two drives — even two hard disks!

Version 4.x speeds DOS interaction for large multiply-keyed files; enables variable length records of virtually any length; verifies accuracy (optionally) with read after write, useful in gritty environments; offers password and data encryption.

There's also Xtrieve, for Btrieve file inquiry and data manipulation, and Rtrieve for report writing. All three in versions for any network that supports the MS-DOS 3.1 file sharing function.

Ask for:	List:	PC Brand:
S0650	**$250**	**$195**
S0652 Network Version	**$595**	**$465**

TERMS AND CONDITIONS OF SALE

Licenses: Each price is for a license to use a product on a single computer and does not constitute its ownership. We will inquire for you about site licenses. Except as otherwise indicated or where "#" follows the Product Code, products may be used to create programs for distribution without royalty payments or additional licenses, provided said programs do not substantially replicate the products themselves.

Compatibility: PC BRAND's standard products are designed to operate with the IBM® PC, XT or AT under PC-DOS and require no more than 128k of RAM unless indicated. Non IBM machines using MS-DOS: contact manufacturer about precise differences so we can advise.

Returns: See box page one. Defective parts will be replaced. Please call for authorization to return a product for refund.

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Brand, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $10, each add'l $4.50. UPS Next Day Air or Federal Express 1-2 Day Air: 1st product $18, each add'l $6. FedEx Next Day 10 AM: 1st product $28, each add'l $7. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

For Orders, Literature, or Catalogs, Call Us at...

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600

PC Brand, 150 5th Ave., New York, N.Y. 10011-4311

Telex: 667962 (SOFT COMM NYK)

Prices, terms, and specifications subject to change without notice.

SAVE $5
Just tell us
this code:
J8

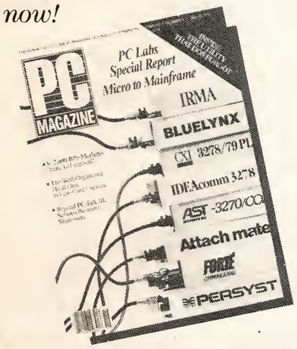
Subscribe to PC Magazine and save 65%

You'll receive in-depth and comprehensive product evaluations and comparative hardware and software evaluations from PC Labs, the publishing industry's *only* testing laboratory.

You'll receive new applications, power tips and special reports designed to add more power and flexibility to yor IBM/MS DOS-standard system.

And you'll network with other users and discover routines and techniques that can save you time and money.

Subscribe to *PC Magazine* today! Send the coupon to: *PC Magazine* · Box 2443 · Boulder, Colorado 80322 *or call Toll-Free, 1-800-852-5200 now!*



Yes, send me PC Magazine for:

☐ 22 issues (one year) for $24.97. Save 62%!
☐ 44 issues for only $44.97. Save 65%!

Savings based on annual single-copy price of $64.90.

MR./MRS./MS. PLEASE PRINT FULL NAME

COMPANY

ADDRESS

CITY STATE ZIP

Add $15 per year for postage outside USA, US currency only. Please allow up to 60 days for delivery of first issue. Basic annual subscription price is $34.97.

8ZD66

ACCELERATED XT & AT

MOTHERBOARD		SYSTEM
$129	8/4.7 MHz	$799

XT-88 Turbo: 8088-2 CPU: BIOS and 8 I/O slots standard, 640K or 256K RAM optional.

$249	8 MHz	$919
$319	10 MHz	$989

XT-186: 80186 CPU: BIOS 1.0 standard, BIOS 2.0 optional, 8 I/O slots: 6 XT type and 2 AT type, 640K or 256K RAM optional.

$629	8 MHz	$1549

XAT 286: 80286 CPU: Phoenix BIOS 6 I/O slots: 3 AT type and 3 XT type. 512 K optional. XT size.

$629	8 MHz	$1649

AT-286: 80286 CPU, Phoenix BIOS, 8 I/O slots: 6 AT type and 2 XT type, 640K or 1M RAM optional. AT size.

Above systems include motherboard plus:

RAM (256K on XT, 512K on XAT, 640K on AT) keyboard & enclosure, floppy drives & controller (dual 360KB on XT, 1.2 MB with hard disk/floppy controller on AT & XAT), monochrome graphic/printer card, power supply. Interface and peripherals optional.

Challenger

CHALLENGER COMPUTER, INC.
122 South Rd., Bedford, MA 01730 TEL: 617-275-3517

NEW DEALER SPECIAL

XT—COMPUTER	QTY 2
640K MOTHER BOARD W/OK	89
256K RAM	27
135 WATT POWER SUPPLY	53
FLOPPY DISK CONTROLLER	27
FLIP TOP BOX	38
XT/AT KEYBOARD	59
MONO/GRAPHICS/PRN (HERC)	54
TTL MONITOR W/SWIVEL	96
(1) 360K FLOPPY	92
PARTS	$535
ASSEMBLY & TEST	64
	$599
20 MEG HARD DISK W/CONTROLLER	425

AT COMPUTER	QTY 2
80286 MOTHER BOARD W 512K RAM	595
200 W POWER SUPPLY	98
AT BOX W/KEY LOCK	79
SERIAL & PARALLEL CARD	65
AD KEYBOARD	59
1.2 M FLOPPY	129
FDD & HDD CONTROLLER CARD	225
PARTS	$1,250
ASSEMBLY & TEST	150
	$1,400
30 MEG 40 MSEC HARD DRIVE	615
EGA CARD	295
EGA MONITOR	449

XT PORTABLE $995

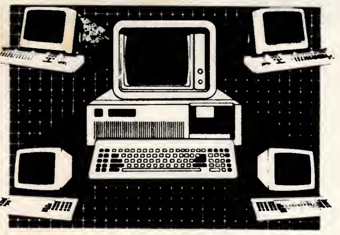
640K MOTHER BOARD
640K RAM
MULTI I/O CARD
CLOCK, CALENDAR,
SERIAL,
PARALLEL
TWO 360K
FLOPPY DRIVES
MONO/GRAPHICS/PRN (HERC)
DETACHED KEYBOARD
135 W POWER SUPPLY
PORTABLE CASE
9" MONITOR, GREEN or AMBER

ADD ON BOARDS — PARTIAL LIST

XT-640K TURBO 4.77/8 MHZ W/OK	99
LOCAL AREA NET CARD W/SOFTWARE	249
384K MULTI-FUNCTION (AST 6 COMP) OK	69
MODEM CARD 300/1200 (HAYES COMP)	149
DISK I/O 2FD CTRL, PRN, SER, CLOCK, CAL, GAME	70
AB SWITCH BOX W/64K BUFFER (1:2 or 2:1)	94
AB SWITCH BOX (1:2 or 2:1)	29
COLOR/GRAPHICS/PRINTER CARD RGB	58

ACS IMPORTERS
5311 DERRY AVE., UNIT A
AGOURA HILLS, CA 91301
HOURS 9 AM - 6 PM PST
(818) 889-1092
TELEX: 299353 POST UR

ATTENTION SOFTWARE DEVELOPERS!
More Users For Less Money!
Digitrol's software solution to multi-user.



pcSHARE
MULTI-USER O/S

pcSHARE allows your IBM-XT/AT or compatible to support up to 5 users running your vertical market programs and 1-2-3, dBASE, WordStar, and more on inexpensive serial CRTs. pcSHARE efficiently runs compiled Basic, dBASE, Pascal & C programs with full DOS 3.0 compatible record locking.

DIGITROL COMPUTERS INC.
440 Phillip Street,
Waterloo Ontario, CAN N2L 5R9
(519) 884-4541

IBM XT/AT, Lotus 1-2-3, dBASE, and WordStar are trademarks of IBM Corp. Lotus Development Corp., Aston-Tate and MicroPro International respectively.

CIRCLE 375 ON READER SERVICE CARD

64K ▪ 128K ▪ 256K
DRAMS
80287-8 ▪ 80287-3
8087-3 ▪ 8087-2
8087-1

BITTNER
ELECTRONICS

899 SOUTH COAST HIGHWAY
LAGUNA BEACH, CA 92651
(714) 497-6200
CALL NOW FOR **FREE** CATALOG

CIRCLE 376 ON READER SERVICE CARD

SUPER LOCK™

IBM PC, XT and AT programmers:

Don't give away results of your effort!
Don't waste your money buying costly fingerprinted diskettes!

You can have your programs automatically protected against piracy!

* **No special training needed!**
* **No confusing instructions!**
* **No discouraging procedures!**

No Sales Tax on orders outside N.Y. State

$ 180

YETIWARE CORP.
P.O. BOX 21152
MID TOWN STATION
NEW YORK N.Y. 10129

DEALER INQUIRIES INVITED

CIRCLE 377 ON READER SERVICE CARD

Protect Your Private Parts.

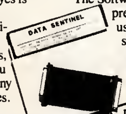
Whether you're a software developer or an end user, whether you work alone on a PC or you're part of a 24-person project team, keeping new software or sensitive data from prying eyes is serious business.

Short of complicated, wizard-like protection schemes, or James Bond, you haven't had too many realistic alternatives. Until now.

Because Rainbow Technologies has developed simple, easy-to-use systems called Sentinels that say no to prying eyes. No matter how hard they look.

The key to Sentinel is just that—a smart piece of plastic smaller than a best-selling cassette that plugs into the parallel port of your IBM PC/XT/AT or compatible.

There are two ways to go. The Software Sentinel™, that prevents anyone from using your licensed software program unless you give them a key. And the Data Sentinel™, Rainbow Technologies' newest prying eye-stopper, that automatically protects sensitive personal or project data as you enter it. The Sentinel Security Systems from Rainbow Technologies. The only hard part is the key.



RAINBOW
TECHNOLOGIES, INC.
17971 Skypark Circle, Suite E, Irvine, CA 92714
Phone (714) 261-0228 Telex: 386078

CIRCLE 378 ON READER SERVICE CARD

PERFECT COPIES GUARANTEED

Why risk duplicating your important programs on your computer, when *our* equipment is designed solely to duplicate disks and verify their perfection 100%?

Over 600 formats. 3½", 5¼" and 8". Plus serialization, copy protection, labeling, packaging, shrink-wrapping and *fast*, personalized service.

Western Transdata Inc.
DISK DUPLICATION

1701 E. Edinger Ave.
Building A4
Santa Ana, California 92705
(714) 547-3383 (Collect)

CIRCLE 379 ON READER SERVICE CARD

Sort RM/COBOL Files
• • • QISORTC • • •

Finally, a sort utility for RM/COBOL. Using efficient algorithms in assembly language making it the fastest sort on the MS-DOS market. Qisortc supports:

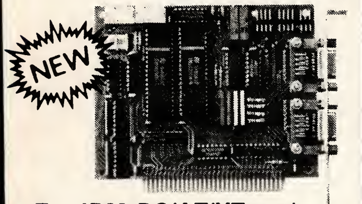
- RM/COBOL Version 1.5x & 2.xx files
  - Single / Dual File Indexed
  - Relative and Sequential
- ASCII, COMP-1, & COMP-3 fields
- 32 sort control fields
  - ascending / descending
- 32 record selection fields
  - EQ:NE:GT:GE:LT:LE field:constant
  - implied AND, OR may be specified
- 32 output fields or total record
- CALL from RM/COBOL, when provided subroutine is linked with runtime
- Use pathnames and return codes
- Keyboard or Parameter file
- Requires MS-DOS ver 2.xx or above
- Only $95.00 - COD, VISA, or M/C.

Quantumn Information Systems, Inc.
145 N.W. 85th St., Suite 103
Seattle, Washington 98117
(206) 789-2888 Telex 4941041

CIRCLE 380 ON READER SERVICE CARD

RS-422
Communications Board

NEW



- **For IBM-PC/AT/XT** and compatibles
- Dual RS-422 serial interface
- Programmable to 56k baud
- Differential drivers to 4000 ft.

$345.00

QUA TECH, INC.
478 E. Exchange St. Akron OH 44304
(216) 434-3154 TLX: 5101012726

CIRCLE 382 ON READER SERVICE CARD

FANSI-CONSOLE™

The Integrated Console Utility™
All the little things IBM forgot!
for IBM—PC, XT, AT & clones.

- 1.2 to 3.0 times faster DOS & BIOS screen writing
- more escape sequences than ANSI.SYS
- usable in any language
- scroll recall facility
- compatibility w/PC & AT software
- full EGA support
- 255 character typeahead buffer
- increase key repeat rate
- no scroll blink for some adaptors
- VT 100/52 emulation
- auto dual screen disable
- keyboard induced breakpoints
- window support
- support for 50 line display
- many many more little features

400p Manual (w/slip case) & disk $75, or just Shareware disk $25. As described in *PC-World* February 86, pg. 282 and in *Lotus* June 85, pg. 8.

HERSEY MICRO CONSULTING, INC.
Box 8276E, Ann Arbor, MI 48107
(313) 994-3259 x363 VISA/MC

CIRCLE 383 ON READER SERVICE CARD

NEW EPROM PROGRAMMER

EPROM 1
$495

*Programs all 28-Pin, 5-Volt EPROMs and EEPROMs.
*Perfect for engineering workstations, field service and small production runs.
*Simple keyboard and display for stand alone duplication.
*Interface to your computer via an RS232C port.
*PC control software included at no extra charge.

TO ORDER PHONE:
800-325-6028 Outside California
916-885-7262 Inside California
VISA AND MASTERCARD ACCEPTED

INTERNATIONAL MICROSYSTEMS INCORPORATED
11554 C Avenue, Auburn, CA 95603

CIRCLE 384 ON READER SERVICE CARD

TECH MART

FREE DISK SPACE

THE FILE STORAGE METHOD USED BY PCTECHLINE.

The ARC file archive utility can save 50% or more on disk storage, and modern transfer time.

- **Not copy protected**
- **Program Sources Included**

Only $50

"A sophisticated and eminently useful product."
—PC Week

SYSTEM ENHANCEMENT ASSOCIATES
21 New Street, Wayne, NJ 07470
(201) 473-5153

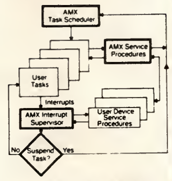
CIRCLE 386 ON READER SERVICE CARD

AMX

8080 Z80 6809 — 8086 8088 — Now 68000

Real-Time Multitasking Executive

- No royalties
- Source code included
- Fault free operation
- Ideal for process control
- Timing control provided
- Low interrupt overhead
- Inter-task messages



AMX is TM of KADAK Products Ltd. CP/M-80 is TM of Digital Research Corp. IBM, PC DOS are TM of IBM Corp.

Options:

- Resource Manager
- Buffer Manager
- Integer Math Library
- Language Interfaces: C, Pascal, PL/M, Fortran
- DOS File Access: CP/M-80, IBM PC DOS

AMX for	8080	$ 800 US
	8086	950
	6809	950
	68000	1600
Manual (specify processor)		75

KADAK Products Ltd.
206–1847 W. Broadway, Vancouver, B.C., Canada V6J 1Y5
(604) 734-2796
Telex: 04-55670

CIRCLE 389 ON READER SERVICE CARD

MYSTIC PASCAL

Fastest Compiler on Earth

NOW ONLY $16!!!

- 640K code, data and stack
- incremental compiler runs in background while you are editing
- effective compiler speed greater than 100,000 lines/min
- immediate execution of Pascal statements
- true multi-tasking for up to 100 Pascal procedures
- full screen editor
- complete Help windows

disk with condensed manual to print out $16—printed manual $16—both $32—for IBM PC 256K Visa/MC/COD—add $2 shipping, COD $5, overseas $10

Mystic Canyon Software
P.O.B. 1010
Pecos, NM 87552 505/757-6344

CIRCLE 387 ON READER SERVICE CARD

ASMLIB

ASSEMBLY PROGRAMMING LIBRARY

- NO ROYALTIES REQUIRED
- Graphics - Color, Herc, Monochrome, EGA
- Floating Point Trigonometry and Arithmetics
- 8087 Supported but not required
- ASCII to Numeric conversions
- Formatted Output (PRINT USING)
- Console I/O with Windowing
- DOS Shells
- Sound Generation
- Plus much, much more

ASMLIB has over 170 functions for the IBM PC/XT/AT or compatible under DOS 2.00 or newer. Supplied in MS Assembly language source code on 3 DOS Diskettes with 215+ page reference manual.

All for only . . . $149.00 ppd

BC ASSOCIATES
13073 Springdale Street, Suite 134
Westminster, California 92683
(714) 741-3015

CIRCLE 390 ON READER SERVICE CARD

the HD test

Professional Test/Format Program For Hard Drives in PC/XT/AT

- Setup interleave, step rate, etc.
- Surface analysis to flag bad tracks
- Load/save setup and bad track files
- Now supports AUTOCONFIG!
- Menu driven, with help windows
- Free PARK program included!
- Order HDTEST today for only $99!

Proto PC inc.
612-644-4660
2424 Territorial Rd., St. Paul, MN 55114
Telex 910-380-7623

CIRCLE 388 ON READER SERVICE CARD

PC TECH JOURNAL

TECH MART TECH BOOK

THE CLASSIFIED ADVERTISER'S FORMULA FOR SUCCESS IN SELLING TO TECHNICAL EXPERTS IN THE IBM MARKETPLACE.

FOR RATES & INFO CALL: (212) 503-5115

PROTECT YOUR COPIES OF PC TECH JOURNAL.

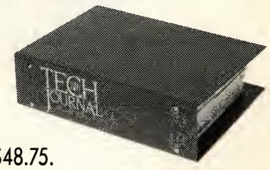
Make your collection of PC TECH JOURNAL a handsome addition to your office or home—and protect and organize them for easy reference!

PC TECH JOURNAL Magazine Binders and cases are made of durable luxury-look leatherette over quality binder board. Custom designed for PC TECH JOURNAL, every order receives FREE transfer foil to mark dates and volume numbers.

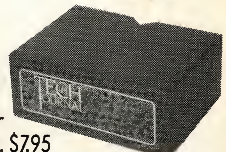
FOR FAST SERVICE CALL TOLL-FREE 1-800-972-5858

MAGAZINE BINDERS

Hold your issues on individual snap-on rods. $8.95 each; 3 for $25.75; 6 for $48.75.



OPEN BACK CASES



Store your copies for individual reference. $7.95 each; 3 for $22.95; 6 for $43.95.

PC TECH JOURNAL
P.O. Box 5120
Philadelphia, PA 19141

Please send ☐ Binders ☐ Cases Quantity ________

Payment enclosed $__________.* Add $1 per order for postage & handling. (Outside USA, add $2.50 per unit ordered, US currency only.)

Charge my:
☐ Amex ☐ Visa ☐ MC (Minimum order $10.)

Card No.________________ Exp. Date________

Mr./Mrs./Ms.________________________
please print full name

Address________________________

City________________________

State________________ Zip________

*PA residents add 6% sales tax.

TECH BOOK

Accessories/Supplies

•• SOFTWARE PUBLISHING ••
GDS offers a wide variety of services that will help get your software to the market. Address your needs with GDS.
- IBM style cloth/vinyl 3-ring binders/slips
- Labels, sleeves, disk pages ...
- Disk duplication with 100% verification
- Bulk diskettes
- Shrink wrapping and assembly
- Quick turnaround

A well-packaged product can make the difference in making a sale. Call us NOW. VISA/MC
Glenco Development Systems
3920 North Ridge Avenue
Arlington Heights, IL 60004
(312) 392-2492

Bar Coding

BAR CODE READERS
- IBM PC/XT, AT, AT&T 6300 & 6300 PLUS keyboard models or RS-232 interface
- NO programming. Reads dot matrix
- Auto-recognition and single code decoding
- Reads Code 39. UPC A/E, Codabar & I 2 of 5
- $635 list, units in stock, 2 year warranty
- Bar code printing software, call for info:

PERCON®

PERCON, Inc.
2190 W. 11th
Eugene, OR 97402
(503) 344-1189

BAR CODE LABEL PRINTING
PrintBar II is a unique memory resident program that prints code 39 and UPC bar codes on Epson, IBM graphics and LaserJet printers. Print directly from almost any program, without copying data to special print files. 30 day money back guarantee. $295 plus $12 S/H second day air.
BEAR ROCK SOFTWARE CO.
P.O. Box 212
Placerville, CA 95667
(916) 622-4640

Business Opportunities

INCORPORATE!
*Save Taxes!
*Protect Your assets!
*Free Booklet!
*Free Forms!
*Save $— Do It Yourself!
*No Legal Fees!
WRITE OR CALL TOLL-FREE TODAY!
HARVARD BUSINESS SERVICES, INC.
3151 Summerset Rd.
Wilmington, DE 19810
800-345-CORP

Personal Computer Owners
CAN EARN $1,000 TO $5,000
monthly selling simple services performed by their computer. Work at home-in spare time. Get free list of 100 best services to offer. Write:
C.I.L.B.Q.
PO Box 60369
San Diego, CA 92106-8369

Hardware/Add-on Boards

ACQUIRE DATA WITH YOUR PC
*16 to 64 differential A-D, up to 8 D-A
*24 to 48 digital I/O, 24 event counters
*Processor and 8 to 40K RAM on board
*Floating-pt processing, Battery backup
*Linearization, 4 selectable gain ranges
*Menu-driven software. Programmable.
*Systems start at $895, 1 year guarantee
WESTERN TELECOMPUTING
202 E. Kagy Bozeman, MT 59715
(406) 586-1511. Or touch-tone, toll-free: 1-800-423-0587, then 328 222.

Z80 Co-processors
For PC and PC/AT. From $199.50. Clock speeds to 9 Mhz. Run CP/M-80 software fast. Develop microprocessor software with our remote debugger kernel. Also run Intel ISIS software tools. We have Z80 processors for VAX and PDP-11 too.
Decmation
2065 Martin Ave #110
Santa Clara, CA 95050
(408) 980-1678

SPEECH SYNTHESIS
SynPhonix: TRUE Unlimited Speech Synthesizer for IBM-PC/XT/AT/jr & compatibles. This low power short card includes an SSi263 speech chip, amplifier and speaker. Software includes Text-to-Speech, Phonetic Editor, Talking Clock & demos. Can be programmed with BASIC and other languages. Prices start below $200.

SynPhonix™
Electronic Speech Articulator

Artic Technologies
1311 N. Main St.
Clawson, MI 48017
(313) 435-4222

RS-232/RS-422 Communications
Versatile communications and I/O boards.
—Four port RS-232 with DOS drivers $349
—RS-422/485-serial interface $149
—RS-232/422 with 24 parallel I/O $199
—RS-422 synchronous with DMA $299
—Delivery from stock
—Custom designs available

adams

Adams Incorporated
P.O. Box 17525
Greenville, SC 29606
(803) 297-9630

FIXED DISK BIOS/BOOT
FiXT boots from most popular Hard Disks—DAVONG, TECMAR, IOMEGA, GT LAKES, etc. Adds XT-like BIOS interface to your disk for PC. Security, multiple volumes, removable media support optional. No-slot plug-in installation. Specify controller and computer with order. $80-$95. Add $3 shpg., CA tax.

GOLDEN BOW SYSTEMS
2870 Fifth Avenue
Suite 201
San Diego, CA 92103
(619) 298-9349

PC-PROMPAC
ROM Expansion for PC!
Aldia systems introduces PC-PROMPAC, a "half-sized" PROM/ROM expansion board for IBM and IBM compatible PCs. PC-PROMPAC will support up to six 28-pin JEDEC compatible devices (ex: 2764, 27128, 27256, 27512, 6264, etc.) with individually selectable address ranges. Prices start at $125 for single units. Quantity discounts and OEM arrangements available. MC/VISA.
ALDIA SYSTEMS, Inc.
P.O. Box 37634
Phoenix, AZ 85069
(602) 886-1786

Hardware/Disk Drives

8 INCH DISKETTE SYSTEM
Read, write and format diskettes from IBM mainframes, minicomputers, data entry equipment, etc. Complete easy to use software handles EBCDIC conversion. Can read and write CP/M 8-inch diskettes (many formats). You also can use 8-inch drives for PCDOS files; 1200 KB per diskette! $1295 complete.
MicroTech Exports
223 Forest Avenue
Palo Alto, CA 94301
(415) 324-9114

Hardware/Peripherals

TURBO START YOUR PC!!!
FOR IBM PC's ONLY!! The PC-INMA a Firmware ROM Module for the IBM-PC!!! Features: BOOT within 10 SECONDS!!! BREAK the 640K BARRIER!! Access up to 704K of RAM!!! (Lotus, DOS, etc.) SCREEN RELIEF-protects your monitor, and MORE!! Easily PLUGS IN-NO SLOT! Only $49.95 + $3 S/H. Dlr. inquiries accepted CK/MO/VISA/MC.
JS&J Software
1281 S. King St. Suite 6A
Honolulu, HI 96814
24 hr Orders Only: (800) 821-5226 ext. 435

IBM PC/XT/AT ...... SPEEDUP
*OVERTRUSTER: Boots the performance of PC and XT by 280%! 100% software compatible, is switchable between 4.77 and 7.38Mhz. $195.
*SPEEDSWITCHER: This is an economical 6 speed switch (4-11Mhz) for AT's. $89
*XCELEX: Automatic 5 speed switch (6-12Mhz) This unit will switch on the fly and works with ALL IBM AT's including the new Models with speed limiting Rom Bios. $149.
DATA MANAGEMENT SYSTEMS, INC.
1525 Hickory St.
Salina, KS 67401
ORDER HOTLINE (913) 823-6440

Rates and Information

Standard listings consist of a bold lead line (25 characters maximum); 7 lines of body copy (45 characters per line); 4 lines for company name, address and telephone number. $140 per insertion—3 issue minimum. Additional charge for extra lines and company logos. Prepayment and frequency discounts available. American Express, MasterCard, Visa accepted. Copy subject to publishers approval. Send typewritten or printed copy, reproducible logo art (if applicable) and remittance to Kathryn Cumberlander, Classified Sales Manager, Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016. For additional information, assistance, or to place an order by phone, call collect (212) 503-5115.

TECH BOOK

Software/Business

The Andsor Collection™
Unique concepts: creates complete, self-contained, window-based data management environment in one DOS file. Simplifies everything. Combine functions to create your own solutions in any application: calculations, database management, modeling, text processing, charts, data analysis, statistics, reports, labels, forms, presentations, mail-merge, etc. Simple enough for a PCjr, sophisticated enough for a PC AT. Superb, 400 page hard-cover manual, with many examples. From simple calculations, files, inquiries, to complex models, data structures, reports: when your favorite data manager/spreadsheet/word processor/integrated system cannot provide the solution you need, remember The Andsor Collection. 60 day money-back guarantee. US$95 + $5 s&h. Visa/MC/AmEx/Chk/MO/COD. Call or write now to order, or for info. IBM/PC/XT/AT/PCjr, 128K+, one drive or hard disk, monochrome and/or color monitor, DOS 2.0+. Not copy-protected.

The Andsor Collection™

Andsor Research Inc.
181 University Avenue, Suite 1202
Toronto, Ontario, Canada M5H 3M7
(416) 364-8423

THE DGI ORGANIZATION—PRINTERS SUPPORTED
Lets you easily manage your organization charts. It is menu-driven for easy use. You can create, save, update and print or plot your organization charts. You can input up to 58 individuals with four lines of text each, two title lines and a footnote. For use with the IBM PC or compatible computer and most printers or HP plotters. $95.
DECISION GRAPHICS, INC.
P.O. Box 2776-PCTJ
Littleton, CO 80161
(303) 796-0341

Software/Communications

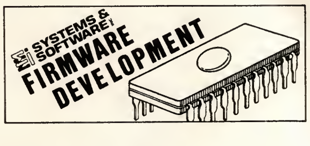
PC SERIAL DATA ANALYZER
Use your IBM PC or compatible to analyze data streams between two serial devices (up to 9600 BAUD). Two windows display each devices transmission in ASCII or HEX. PC can also act as a terminal for either device. Invaluable tool for debugging serial interfaces. Disk & manual $150.
Triple C Software
2897 SW 13th St.
Fort Lauderdale, FL 33312
(305) 583-0687

Software/Development Tools

ROMable CODE on PC!
PCLOCATE allows PC users to develop ROM-based software from MS-DOS "Exe" files. The user specifies the physical location of all segments. Output files are compatible with most PROM programmers. PCLOCATE supports the 8086, 8088, 80186, 80188, and 80286 processors. MC/VISA.
ALDIA SYSTEMS INCORPORATED
P.O. Box 37634
Phoenix, AZ 85069
(602) 866-1786

REALIA COBOL SITES
REALFORM is the Screen Handler you have been looking for. It features interactive screen design, an on-line forms editor, relief of cursor control, expedites screen generation, supports numerous data field types, is easy to use and offers powerful data entry functionality. $395. US.
CHARANDON COMPUTER SYSTEMS INC.
1600 Laperriere Avenue
Ottawa, Ontario, Canada K1Z 8P5
(613) 729-3003

FIRMWARE PRODUCTION ON PC
LINK&LOCATE enables PC users to produce ROM-based firmware for 8086/87/186 from object files generated by C, PL/M compilers & MASM. Provides full control of segments placement anywhere in memory. Supports output of INTEL hex file for PROM programmer, absolute object file for symbolic debugger & ICE, and MS-DOS EXE fifle. Includes an INTEL compatible linker, locator, librarian and hex formatters. $350.

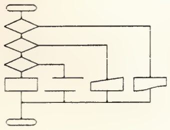


Systems & Software, Inc.
3303 Harbor Blvd., C11
Costa Mesa, CA 92626
(714) 241-8650

C-INDEX
C-INDEX is a state-of-the-art data management library for C. Designed for serious product development. No Royalties. B+Tree, variable length records, multiple keys, and transportable source code. Call for our free info packet.
C-INDEX/PRO $195 Partial Source
C-INDEX/PLUS $395 Full Source
TRIO SYSTEMS
2210 Wilshire Blvd. Suite 289
Santa Monica, CA 90403
213-394-0796

GENSCREEN FOR MS-COBOL
Cobol Source Code Generator for generating the screen section and data division cobol source code for Microsoft and IBM PC cobol. Screen Image Text files are run through GENSCREEN to produce all of the source code for your screen in less than a minute. Super fast programmer productivity tool $69.99.
Personal Computer Development Corporation
P.O. Box 8556
Warwick, R.I. 02888-8556
(401) 333-8704

FLOWCHARTS BY PROFLO
The fast, friendly, flowchart drawing program for the software professional. PROFLO speeds software development and reduces documentation costs. Lets you create, edit, and print clear, precise flowcharts, quickly and easily. Requires IBM PC (or clone) with DOS, 128K, graphics monitor & graphics printer. Program & Documentation $49.95. Demo Disk $10.00.



Star Systems
28 Topstone Drive
Bethel, CT 06801
(203) 744-0546

TURBO PASCAL GENERATOR
GTP APPLICATION DEVELOPMENT SYSTEM, version 2. Builds complete, working application systems. You give it spec's, it writes error-free code.

Indexed Data Bases	Multiple Screens
Context Sensitive Help	Memory-mapped Video
Global/Phonetic Search	Full Keyboard Supt

Easy to Use Price $150.00 Visa/MC, ck, MO
AEF
P.O. Box 928
Katy, TX 77492
(713) 391-8570

SCREENWARE UTILITIES
FAST ELEGANT SCREEN CREATIONS FOR PC ASSEMBLY ROUTINES. Make new or edit existing screens, even those CAPTURED in OTHER pgms BY OUR PGM! Supports ALL color/char/att & monitors. Pop-up menus, help & file management. Creates binary & assembler-ready ASCII source files. Easy to link to main pgm & invokes w/little code. KBD template, charts, 50pg Man, PC/AT-128K, ONLY $27!
Crosswinds Software
8621 Windjammer Drive, Suite M-56
Raleigh, NC 27609
(919) 847-1812

ADVANCED MODULA-2 TOOLS
REPERTOIRE: $64. includes 360K of Logitech or ITC Modula-2 source code: screen compiler and fast display system with input checking, help, windows, menus, colors; auto-adapts to hardware. Integrated multiwindow editor; natural-language analyzer. DOS & BIOS tools. FREE documentation & demo disk. MC/VISA
PMI
4536 SE 50th
Portland, OR 97206 (503) 293-7706
MCI Mail: PMI; Compuserve: 74706,262

WORK HORSE BY USERSOFT
The BASIC code generator that provides:
1) screen painting 2) help screens
3) structured programs 4) maintainable code
5) cuts programming time 6) INKEY$ routine
7) merging 8) printing 9) cross-references
10) runs interpretively or can be compiled.
All this and more for just $59.00.
USERSOFT, Inc.
P.O. Box 517
Manteo, NC 27954
(919) 473-1970

THE PL/M CONNECTION
- Develop PC software with your PL/M compiler.
- Complete Interface to PC-DOS and PC-BIOS
- Fast Graphics Routines example programs
- All source files are included
- 200 page Technical Reference Manual
- $295.00 No Royalties.

CompuFirm Corporation
7841 Balboa Ave #108
San Diego, CA 92111
(619) 571-0228

BASIC PROGRAMMERS
ADD MORE HORSEPOWER TO YOUR PROGRAMS!! Use THE BASIC ENGINE™ to build YOUR programs in FAST compiled BASIC. Add your modules to a working core of assembly routines, menus, data screens, pop-up help, sort, more; over 60 subroutines. Complete source for two working programs, full instructions, $79.95. Requires DOS 2+, MS-QuickBASIC compiler.
Praxis Software Engineering Associates
370 Andrews St.
Green River, WY 82935-4846
(307) 875-9626

PRODUCTIVITY TOOLS
SRMS™ Software Revision Management System stores all versions of source code in a single library. Allows retrieval of any version of source and application of changes while recording when, why, and where changes were made with no duplication of common code. DOS pathname, directory, and environment variable support, typeset manual, much more.
New version (2.0)........................ $125.00.
QMAKE™ is an intelligent system builder patterned after the UNIX make utility. Only compiles those routines that have changed since last build. Support for macros, multiple entry points, command line parameters. Integrates fully with SRMS™........................ $99.00.
MS/PC-DOS 2.0 (MN plus 6%) MC/VISA
QUILT™ COMPUTING
7048 Stratford Rd.
Woodbury, Minnesota 55125
(612) 739-4560

Better BASIC Programmers
BetterTOOLS 2.00 SPEEDS BetterBASIC 1.1/2.0 development. Includes: 190+ useful tools in 17 modules, manual, and source code. No royalties. Quicksort, screen builder, extended math, disk directories, display and printer routines, powerful input, data encryption, on-line error descriptions, much more. Only $89. VISA, MC, COD.
SOFTWARE ASSOCIATES
6220 W. Airport Blvd.
Houston, TX 77035
(713) 726-0706

PC CROSS-ASSEMBLERS
Up to 10,000 lines per minute! Fast X-ref and Linker plus Macros and Librarian. Generates HEX, TEKHEK, S-records, and .OBJ output records. Over 40 micros and XENIX, MS DOS, CPM 80 and ISIS versions. Accepts MOTOROLA and INTEL directives and Mnemonics.
RELMS™
P.O. Box 6719
San Jose, CA 95150
(408) 265-5411

TECH BOOK

SUPERTECH "PREP" $19.95
Powerful, exceptionally flexible, language-independent pre-processor. Use it, for example, to write register, opcode and operator synonyms, nested macros, line-length identifiers, etc. with assemblers without such features built-in. OK to share. Phone 24 hrs. (Please add $3 s/h, WA res. add 7.9% tax)
SUPERTECH
11410 NE 124 St. Suite 6143
Kirkland, WA 98034-4399
(206) 488-9253

SOURCE CODE LIBRARIAN
TLIB™ stores ALL versions of your source in ONE compact library file, even with hundreds of revisions. Updates (deltas), 5-7 times faster than Unix SCCS. Date & comments for each version, easy retrieval. LAN-shared libraries. "Free public domain MAKE (with source) by Landon Dyer." DOS 2.x/3.x $99.95 $3 s/h VISA/MC.
Burton Systems Software
P.O. Box 4156
Cary, NC 27511-4156
(919) 469-3068

LAZY ASSEMBLE
Batch program assembles/links multiple .ASM's. Specify which error-free .LST files you want printed. Delete form feeds from .LSTs' if desired. Link only if all assemblies ok. Other utilities: link map re-arranger and case sens/insens high speed string search across editable file list. $49.95 (CK/M.O.)
Goldbrick Software
1376 Deerport Ave. #22
Fullerton, CA 92631
(714) 524-6604

Fortran Addenda '86
Libraries for graphics and friendly/interactive programs. ASMUTIL2: Total PC control; printers (3), CRTs (2), disks, FULL keyboard, strings, high-speed gets/puts, line/box, fills, tile-painting, EGA graphics modes. BUTILE2: Input word-processing/editing, non-overflowing formats, window management, ... 100 easy to program, "smart" routines + defaults/toggles. 170 pg. manual & annotated samples. $95 alone; both $165. Specify compiler and version.
IMPULSE Engineering, B.R. Strong, Jr.
P.O. Box 3540
San Francisco, CA 94119-3540
(415) 788-4611

The SCREEN GENERATOR
SAVE TIME! Powerful Screen Designer and Memory Resident Screen Manager eliminates tedious screen management tasks from your program. Handles all screen display & data entry. Easy access to Screen Manager from your program. Not a Code Generator! NO Royalties, Not Copy Protected. Easy conversion from other systems. Use with BASIC, TURBO ($59), FORTRAN, C, Realia COBOL, BetterBASIC, PL/M-86, or Assembler. Call if not listed. Price $125.
THE WEST CHESTER GROUP
P.O. Box 1304
West Chester, PA 19380
(215) 644-4206

TREES END DIAGRAM LAYOUT DRUDGERY $29.95!
Decision trees; composition of data flows and date stores; Functional decomposition; Family trees, Process descriptions and structure diagram; Structured English; Warner-Orr data structure, process structure, and data flow. TREES frees you from positioning symbols!
- Fits brackets as your text grows/shrinks.
- Line wraps and scrolls each text element.
- Diagram stores as machine-readable tree.
- 1000 elements + 32760 characters/diagram.
- Requires MS-DOS, hard disk, 512K.

Ver. 2.0 $29.95 VISA/MC orders: 618-242-4161

Trees™
Box 687
Mt. Vernon, IL
62864

StruBAS- the BEST BASIC!
Adds structure, full-screen handling, indexed files and menus to BASIC. Translates structured BASIC to Microsoft BASIC. Provides interactive screen definition and creates runtime code and tables. Includes subroutine library and sample application. Not copy protected. $150 + $5 s/h. Demo $5. VISA/MC
LANEY SYSTEMS, INC.
3 Office Park Dr. Suite 100
Little Rock, AR 72211
501-225-7755

Documentation Generator
Customizing Documentation Generator offers:
FLEXIBILITY—define your documentation
SIMPLICITY—follow DOS tree structure
CREATIVITY—run all commands and programs
COMMAND ABILITY—repeat an expanded command
I/O ABILITY—run special I/O redirection
check $39, MO, VISA/MC.
U-NEED & SEE COMPUTER SYSTEMS
P.O. Box 7363
Thousand Oaks, CA 91359
(805) 495-8441/(818) 889-3324

METRICS FOR MANAGEMENT
ANALYZE reads your code and produces metrics on VOLUME, COMPLEXITY, PRODUCTIVITY & more! Management, QA/QC & Engineers gain valuable insight into scheduling, problem areas, testing, maintenance & compliance to standards. For IBM PC. Process FORTRAN & "C" $495.00 + S/H. Add Ada for only $200.00 more! MC/VISA .
AUTOMETRIC INC.
891 Elkridge Landing RD., Suite 350
Linthicum, MD 21090 (301) 859-4111

SOFTWARE/ ENGINEERING

METAL FABRICATORS
PC/Cutlist takes input from your bill of material—Detail drawing and calculates the best cutting combination for any length stock and prints a shop ready cutting list and scrap report. Also an optimization feature finds best mult length for mill orders. Price $300. Demo Disk $25.00
THE JOSEPH ALBERT CO.
P.O. Box 611
Blue Island, Illinois 60406
(312) 349-9032

CREATE PCB's EASILY
Easily create double-sided printed circuit board artwork of up to 21" square using PC-Layout on an IBM-PC. the artwork is created using a color artwork editor. the final artwork, solder mask and silk-screen (text) is produced using either a printer or a plotter. Price $745. Demo disk/manual $45. Demo disk credited to final purchase.
Draco Technology
7210 Jordan Ave., Suite D50
Canoga Park, CA 91303
(800) 235-6646 ext. 646 (outside CA)
(800) 235-6647 ext. 646 (CA only)

PC/8087 VECTOR PROCESSING
The VECTOR87 library is written in assembler, includes 60 routines to turbocharge your math & scientific programs. FFT, vector operations, convolution, etc. 1K real FFT takes only 1.2 second. Versions for Fortran (MS, R/M, Lahey), C (MS, Lattice). $120 each version with source, no royalties. Write for technical information.
VECTORPLEX Data Systems Ltd.
P.O. Box 138, Station M
Calgary, Alberta, Canada T2P 2H6
(403) 248-1250

SOFTWARE/ FILE MANAGEMENT

Wilkes Database
For IBM-PC and compatibles. Paint screens to create/view/update/delete/sort/search/print/reformat/arithmetically manipulate new or extant files. ASCII, 16/32 bit integer and 32/64 bit floating point storage. Powerful key sort and report formatting. $325. Demo $40.
Wilkes Software
5231 Longwood Drive
Memphis, TN 38134
(901) 386-4712

SOFTWARE/FINANCIAL

FINANCIAL CALCULATOR
Memory resident software for PC/XT/AT or compatible computers. Non-resident version also available. Loan amortization, internal rate of return, IRA, real estate, savings, depreciation, and much more!!! FREE shipping in the U.S. 30 day money back guarantee.
NOT COPY PROTECTED. $49.
OPUS Engineering
5825 Vargas Ct.
San Jose, CA 95120
(408) 997-1503

SOFTWARE/GENERAL

PUBLIC DOMAIN SOFTWARE IN C
Over 60 volumes of public domain software in CP/M & MS-DOS formats.
- editors & compilers
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details.

C Users' Group

THE C USERS' GROUP
P.O. Box 97
McPherson, KS 67460
(316) 241-1065

SOFTWARE/GRAPHICS

TEKTRONIX 4010 EMULATION
High resolution screen/printer graphics for the IBM PC. Full interactive capabilities with file transfer and cross-hair control. Utilizes PLOT 10, protocols. OFF-line review of graphics output. Supports IBM color/graphic or Hercules cards. Easy to use 4010 emulation at an affordable price. $80.00. Demo disk $5.
Technological Systems Group
5044 Haley Court
Lilburn, GA 30247
(404) 923-4980

SCIENTIFIC DATA PLOTTING
SCI-GRAF creates graphs up to 1680 x 1712 dots (over 3 million pixels!) on Epson or IBM graphics printers. Supports log scaling, overlays, point-labeling, legend creation, batch mode, wide-carriage printers, and color graphs on a JX-80. Requires DOS 2 or 3, 256K. No credit cards. $99.95
Microcomputer Systems Consultants (MSC)
27-P Forest Avenue
Port Jefferson Station, NY 11776
(516) 928-7493

TECH BOOK

FORTRAN GRAPHICS LIBRARY
MicroGlyph's SciPlot provides MicroSoft or IBM Professional FORTRAN graphics library on IBM PC/XT/AT under PC DOS. Features high resolution plots from printers (IBM/EPSON/Apple LaserWriter), or graphics displays (IBM Color/HERCULES-Monochrome/TECMAR-GM). Price: $69.95 + $3 shipping + 5% MASS tax.
MicroGlyph Systems
P.O. Box 1066
East Arlington, MA 02174-9998

COMPUTER AIDED GRAPHICS
Free form 2D full screen graphics editor for the IBM-PC and compatibles. Use it to create charts such as: Organization, Gantt, Flow and Pie. Also produces Graphs, Work area plans, Electronic schematics and more. Drives HP or compatible plotters. Only $49.95 a price of a game. For more details send a long SASE to:
WorkWare
P.O. Box 661295
Sacramento, CA 95866-1295
(916) 922-5819

MetaWINDOW™/TurboWINDOW™
Advanced graphics toolkit provides Xerox Star/Apple Macintosh style graphics on your IBM PC. Supports most popular graphics cards. Allows you to create pop-up menus, windows & icons; use proportionally spaced fonts; rubberband & drag lines, text or bitmap images; supports mouse-cursor tracking. Tightly optimized for use with Turbo Pascal, IBM Pascal, C, Fortran.
METAGRAPHICS SOFTWARE CORP.
4575 Scotts Valley Drive
Scotts Valley, CA 95066
(408) 438-1550

FORTRAN GRAPHICS LIBRARY
GRAFMATIC (screen graphics): 75 MS FORTRAN/Pascal, R-M/Profort, Lahey FORTRAN callable subroutines. Fully documented, prof. graphics capabilities, inc. general utility, 2-D interactive, total 2-D plots, 3-D plots and solid models. $135. H-P or H-I plotter? get PLOTMATIC, complete plotter graphics library. Interfaces w/GRAFMATIC. $135. Both $240.
MICROCOMPATIBLES, INC.
301 Prelude Drive Dept. J
Silver Spring, MD 20901
(301) 593-0683

SOFTWARE/LANGUAGES

EASY WAY TO MASTER DOS!
New handbook shows how, why, and when to use all 44 PC & MS DOS commands quickly and easily! Lively hands-on tutorial makes learning fun, for a change. Covers basic thru advanced topics. All versions up to 3.1. Lots of practical examples. 221 pages only $15.95 at bookstores, or send check + $1.00 S&H to:
LITTLE, BROWN & COMPANY PUBLISHERS
200 West Street, Code T-368121
Waltham, MA 02254
1-800-343-9204 (credit cards)

SOFTWARE/NETWORKING

LAN SOFTWARE
LAN software for your PC, XT or AT, allows you to connect to your VAX and other systems. Provides file transfer and terminal services at hundreds of kilobits/sec. TCP/IP or XNS protocols available; MS-DOS, XENIX, VENIX and PC/IX supported, many additional options, basic PC package only $350.

Network Research Corporation

NETWORK RESEARCH CORP.
2380 North Rose Avenue
Oxnard, CA 93030
(805) 485-2700, TLX 297579

SOFTWARE/ONE OF A KIND

INTRODUCING-JORJ
JORJ- The first and only phonetic dictionary! 50K words/15K definitions. Finds words super fast for both good/poor spellers. JOSE a fun word-game is included in iinitial release as a bonus. Requires IBM-PC DOS2 & Hard disk. Copy protected. Price $100. JORJ demo disk $10. Check/M.O. Mich. residents add 4% sales tax.
JORJ SOFTWARE CO.
4354 Fletcher Rd.
Manchester, MI 48158
(313) 428-8010

SOFTWARE/OPERATING SYSTEMS

UN/EMULATOR BY WAWA
Run CPM on your PC or clone at MACHINE SPEED. UN/EMULATOR runs native 8080 code and increases DOS performance by 10%. We include disk conversion software, terminal emulators, and serial transfer utility. Available in 5 MHz and 8 MHz. Starting at $99.95 plus $5 S&H. Dealer inquiries welcome. Call or write:
WEITZMAN AND WOOD ASSOCIATES
580 N.W. 99th Way
Pembroke Pines, FL 33024
(305) 432-8025

SOFTWARE/PUBLIC DOMAIN

TURBO PASCAL™ SOFTWARE $6
Write or call for information about:
- Systems & applications development tools
- Programs for home and business
- Communication tools & applications
- Games in specialized applications
- Scientific/engineering programs & routines
- Graphics including animation tools

TURBO S.I.X.
P.O. Box 8373
Waco, TX 76714
(817) 753-2182

THE BEST OF THE BEST!
Public Domain & User Supported Software for IBM PC & Compatibles! Wordprocessing, Accounting, Spreadsheets, Database, Modem, Games, Languages, etc., etc. 50 disks crammed Full-$205.00! or rent for 2wks. $75.00. Info. and Super Sampler Disk $6.50. Deluxe Word Processor $6.50. Both $12.00. MC/VISA.
BLUE CIRCLE GROUP, Inc.
P.O. Box 23502; Dept. TJ
Minneapolis, MN 55423
(612) 823-4111

SOFTWARE/SCIENTIFIC

SPARSE/DENSE MATRIX
MATRIX: Does to matrix what TK! solver did for numbers. Solves lin/nonlin/diff eqns. Programmable Matrix functions. $59.95
SPARSEPACK: Turbo Pascal source code for user-transparent Sparse Matrix handling. Overcome 64K barrier and more. $49.95. VISA/MC call 1-800-835-2246 ext. 80.
SoftTech Inc.
18505 W. 8 mile.
Detroit, MI 48219
1-313-544-8544

SCI/ENG GRAPHICS
OMNIPLOT [S] (screen graphics) & OMNIPLOT [P] (plotter driver) provide integrated engineering/scientific 2-D & 3-D graphics with NO PROGRAMMING! Menu-driven, flexible, professional. Choice of formats: tabular/line, contour, bar, pie, 3-D wire frame & much more! OMNIPLOT [S] $195. Add OMNIPLOT [P], both $295.
MICROCOMPATIBLES, INC.
301 Prelude Dr. Dept. J
Silver Spring, MD 20901
(301) 593-0683

SOFTWARE/SECURITY

"NEW" BIT-LOCK® SECURITY
Piracy SURVIVAL 4 YEARS proves effectiveness of powerful multilayered security. Uses rapid decryption algorithms and small reliable port for transparent security device. NOW AVAILABLE for PARALLEL or SERIAL port. NEW KEY-LOK™ security device available at HALF-PRICE.
MICROCOMPUTER APPLICATIONS
7805 S. Windermere Circle
Littleton, CO 80120
(303) 798-7683 or 922-6410

SECURE AT/XT/PC
Control system access, data access! FiXT/S. Control system boot for most popular XT/PC hard disk controllers. Feature for AT-and-XT-compatible HD controllers segments hard disk by volumes, controls access with passwords, supports hard disk expansion. $80 $120 + $3 shpg. plus CA tax.

Golden Bow Systems
2870 Fifth Ave. Suite 201
San Diego, CA 92103
(619) 298-9349

SMART COPY PROTECTION
In the past, Copy-Protecting your programs meant inconvenience to the end-user, limited or no hard disk support, and a lot of added expense per product copy. Our new software based EVERLOCK System solves these problems. For IBM and compatibles, call for info.

AZ

Az-Tech Software Inc.
423 Grandview
Richmond, MO 64085
(816) 776-8153

SOFTWARE/SERVICES

TAPE/DISK CONVERSIONS
Conversion services to or from over 500 computer systems:
- Magtapes
- Micro Computers
- Mini Computers
- Word Processors
- Typesetters

Our conversion capabilities surpass most in the industry.
Pivar Computing Services, Inc.
165 Arlington Hgts. Rd. #T
Buffalo Grove, IL 60089
(312) 459-6010

RENT POPULAR SOFTWARE
Finally, you can rent popular software for your IBM or compatible. See how good that game really is. Try that business package before dishing out huge amounts of money. Not public domain. No membership fee. Where else can you get $75.00 programs for $5.00? Send $1.00 for complete catalog.
R.S.D.
Dept. TJ
P.O. Box 272
Bronxville, NY 10708

TECH BOOK

Software/Statistics

RATS!
RATS is the ultimate econometric software package. It is powerful, fast, accurate, and inexpensive. RATS performs regression analysis, time-series forecasting (including Box-Jenkins) and cross-section (including logit and probit). RATS supports a variety of data formats, including Lotus WKS files. RATS also offers a programming language that provides maximum flexibility and power. Spectral analysis and graphics are available as options. Base program $200. Demo including 300 page manual $40. MC/VISA.
VAR Econometrics
P.O. Box 19334
Minneapolis, MN 55419
(612) 822-9690

SW/STATISTICS
TWG/ARIMA—a univariate Box-Jenkins forecasting package, designed for statisticians.
EASI/ARIMA—same as above, for the non-statistician. $450. each.
ELF—The Statistical Package—a general purpose statistical package. $350. Call or write for more information.
THE WINCHENDON GROUP, INC.
P.O. Box 10339
Alexandria, VA 22310
(703) 960-2587

Software/Terminal Emulation

Color VT 102 & 4010
PC102 turns an IBM PC/XT/AT/jr to a VT 102/52 terminal with printer/ASCII & KERMIT file transfer support. Optional: 132 col. 4010 graphics support. Guaranteed compatibility with all VT 100 applications including EDT, WORD11, All-In-One, & UNIX. Easy to use. From $89. Call for free package!
General Micro Systems
P.O. Box 5330, Dept. TJ
Hopkins, MN 55343
(612) 944-0593

SIDETERM
Pop-up VT200/100/52 emulation package accessible from inside any program. Window can be moved and sized at any time. Data can be cut/pasted between terminal and application. Virtual 132 col. screen. Supports EDT, UNIX, and scrolling regions. Not copy protected. $59.99. CK/MO. Demo $10. MO res. add 6.225% tax.
SOFTWARE INNOVATIONS
P.O. Box 32466
Kansas City, MO 64111
(816) 756-2745

Software/Utilities

AT/XT/PC HARD DISK EXPANSION
"Replace hard disk with a bigger one, or add a second drive! Vfeature BREAKS THE 33 MBYTE BARRIER on standard AT, XT, and compatible hard disk controllers. Includes multiple volumes, security features, selectable clusters, keyboard lock. $80-$120 + $3 shipping + CA Tax"

Golden Bow Systems
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

DISK MECHANIC
THE ULTIMATE Floppy Disk Backup & Repair Utility. Can back up ALL software protected disks written on the IBM PC. Works manually or automatically. Files or sectors can be restored, searched, examined & changed. Checks disk drive speed, req. IBM PC, XT, AT, DOS 1, 2, 3, 192K + 64K if only 1 floppy drive. $73 ppd. USA
MLI MICROSYSTEMS
PO BOX 825, Dept TB2
Framingham, MA 01701 USA
(617) 926-2055 for info MC/VISA

CHARACTER CUSTOMIZATION
CHARGENI 2.1 works with the IBM EGA to let you modify the character set, allowing many word processors to display technical material, equations or other special characters. Also works on the Color/Graphics Adapter in graphics modes. Requres DOS 2.x or 3.x, IBM Standard or Enhanced Graphics Adapter. $35 + $2 s/h (MN add 6%).
DK Micro Consultants
P.O. Box 6714
Minneapolis, MN 55406
(612) 722-0931

TallScreen—DOS POWER
Natural extension of DOS. Scroll back through screen output, edit text on full screen, mark blocks to printer or file, recall commands & directories, enter multiple commands, capture screens from application programs, create user profiles. Solid tech support. A real bargain at $49.95 VISA/MC

QUALITAS™

Qualitas, Inc.
8314 Thoreau Drive
Bethesda, MD 20817
(301) 469-8848

Directory eXtended™
BE IN COMMAND! New version of very popular command-oriented file manager! POWERFUL batch file generation-perform any task! FLEXIBLE file matching and sorting! CONTROL file attributes! TEST file integrity! MONITOR disk params; 2-col display! MUCH MORE! Send $35+$5 p&h (in TX, add tax); money back guarantee!

DIRECTORY
eXTENDED(TM)

ECONO-SOFT Dept. T66
9200 North Plaza # 1906
Austin, TX 78753

HELL uva SHELL
Most DOS interfaces are designed for IDIOTS! Get the BEST full-screen interactive directory around ... DOS command recall ... command aliasing ... multiple commands at DOS prompt ... find files anywhere on disk ... find complex text patterns in files ... browse files ... copy files based on time stamp & MORE. Only $39! DOS 2.0+, uses 96K memory. NCP!
Dwarf Nebula Software
666-SHELL
P.O. Box 46, Dept. TJ
Sugarland, TX 77487

EASYINDXE PLUS™
Cross reference all source code by file name. Create index by page number. Unlimited number of indexes and accessed files. Word or phrase indexes up to 59 characters. Table of contents generation, typesetting, and print formatting. Supports over 100 printers. $39.95 $3 shipping (CA. RES. ADD $2.60 Tax)
Foreman Software
977 Clayton St. D4
San Francisco, CA 94117
(415) 665-6414

TurboRef 4.0
Pascal programmers get organized with TurboRef! TurboRef will cross reference a source program and create a program listing. TurboRef can process a list of files, will read "include" files, identifies line number for each reference. New release now avail. IBM PC and jr., XT, AT or compatible, only $49.95; VISA/MC, or check.

GRACON SERVICES, INC.

GRACON Services Inc.
4632 Okemos Rd.
Okemos, MI 48864
(517) 349-4900

TURBO PACKAGE for Turbo Pascal
*Crash the 64K barrier. Forget MODULA2.
*1000 times faster than chains or overlays.
*Precompiler turns short calls into long calls. Run 500K of memory resident code.
*32 bit integer math toolbox, ASM.
*Mix Turbo, TurboBCD, and Turbo-87.
*Free information.
*Turbo Package plus tools, $39.95.
CONVERSATIONAL COMPUTER SYSTEMS
5371 Verbena Road
San Antonio, TX 78240
(512) 692-0353

LIST-IT
Make your IBM BASICA and IBM compiled Basic program readable with a pretty print. Cross references for statements and variables are included. While-wend, FOR-NEXT and IF-THEN-ELSE structures recognized. $29.95. Calif. residents add 6%. CHECK/MO/VISA/MC
PCS
50 Cornell #9
Irvine, CA 92715
(714) 854-1282

SAVE THAT SCREEN!
Do you immediately reach for the PrtSc key to save screen info? What a waste of time and paper! Now, SCREENSNAP™ lets you save and recall up to 9 screens at the touch of a key. Friendly with other resident programs but unlike some it is compact; will run in as little as 5K. Also includes useful utilities to save and recall from files, programmer's interface and sample code. Build your own help screens with your text editor, then save and recall them with SCREENSNAP. $39.
Aldridge Real-Time Software Inc.
Call 800-443-4160; NJ (201) 846-7242

REPEAT PERFORMANCE™
OUTSMARTS IBM PC KEYBOARD LIMITATIONS!® Smoothly ACCELERATES CURSOR MOVEMENT, speeds up editing. High speed scrolling. You set ADJUSTABLE REPEAT SPEED and DELAY. SKID SQUELCH™ feature auto-adjusts to match performance of word-processors/text editors/spreadsheets. Cursor & scrolling STOP INSTANTLY when key is released. EXPANDS TYPE-AHEAD BUFFER-type DOS comamnds while your programs compile. Less than 2K resident. WARNING: ADDICTIVE! IBM/PC/XT/AT compatibles. $back guar. $39.95 +$5 p&h. VISA/MC/AMEX/CK. Site licensing from second copy!

popular demand™

Popular Demand Inc.
62 South 1025 East
Linden, UT 84062
(801) 785-0101

FILE PRINT MANAGER
GLISTER™
★ Use DOS wildcards to build a list of up to 100 files to print
★ Save/restore file lists
★ Restart a file on any page after a printer jam
★ Print multiple copies
★ Control: margins, line/page length, spacing, user-formatted header/footer lines and more
★ Prints files as fast as printer is capable $49
Aldridge Real-Time Software Inc.
Call 800-443-4160; NJ (201) 846-7242

TECH BOOK

AUTOMENU™ VERSION 4.0

Create one menu system to run all your programs, batch files and DOS commands. "Insulates" novices; many options for power users. On-screen help, password protection, user-defined prompts. Written in assembler. 16K size. Over 7,000 satisfied users. Money back guarantee. $46 + $4 s/h. Chk/Visa/MC.

MAGEE ENTERPRISES

Magee Enterprises
6577 Peachtree Industrial Blvd., Dept. T8
Norcross, GA 30092-3796/USA
(404) 446-6611

DISK UPGRADE BIOS for ATs

DUB-14 overides AT Drives Table to allow any compatible drive to be attached and fully used on the standard AT controller. Two ROMs plug into empty sockets on system board. Includes complete Set-Up routine and low-level format facility. Works with UNIX, XENIX, other OS and networks. $95 + $3 shpg. CA tax.

GOLDEN BOW SYSTEMS
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

HARD DISK EXPANSION!

Disk Manager allows the installation of any ST506 hard disk on PC,XT,AT and compatibles. Volumes up to 256mb! Menu driven/auto install, compatible w/ all vers of MS/PC DOS (does not modify DOS), up to 16 volumes, easy to use! $125+ ship. Ask about Novell product! Dealer inquiries invited.

ONTRACK COMPUTER SYSTEMS INC.

Ontrack Computer Systems, Inc.
7460 Washington Ave. S.
Eden Prairie, MN 55344
(612) 941-4504

VCACHE GETS YOUR DISK MOVING!

Hard disk accelerator increases speed of cartridge and fixed disk operations using memory caching to eliminate repetitive disk access. Allocate up to 15Mb of extended or expanded memory, or .5Mb of standard memory for caching disk data. Includes diskette and screen accelerator modules. Automatic and transparent after installation. $65+ $3 shpg, CA tax.

GOLDEN BOW SYSTEMS
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

SOFTWARE/WORD PROCESSING

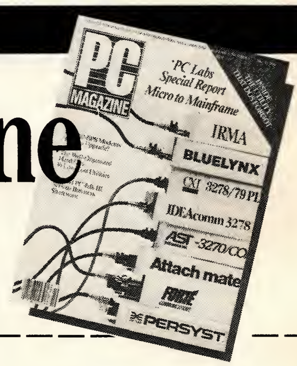
FORMATH™ TEXT-FORMATTER

For technical texts. Sizes & places equations, matrices, ratios, integrals, big symbols, footnotes. Macros; multiple fonts, 158 Greek/math symbols. Automatic hyphenation, section/equation/reference numbering, indexes, table of contents. Dot-matrix, daisywheel, laser printers, plain/graphics monitors. $400. $50 for demo.

$$\lim_{n\to\infty}\sum_{i=1}^{n}\begin{bmatrix}\frac{1}{i(i+1)} & \frac{1}{2^i}\\ \frac{(-1)^i}{2^i} & \frac{1}{\pi^2 i^2}\end{bmatrix} = \frac{1}{2}\begin{bmatrix}\int_0^1 \frac{\log(1-x)}{-\pi^2 x}\,dx & -1\\ \frac{1}{3} & 1\end{bmatrix}^{-1}$$

SHANTHA SOFTWARE, INC.
50 West 97th St. Room 11N
New York City 10025
212-222-7647
Touch-tone Toll-free: 950-1088-wait-FORMATH

Subscribe to PC Magazine and save 65%



You'll receive in-depth and comprehensive product evaluations and comparative hardware and software evaluations from PC Labs, the publishing industry's *only* testing laboratory.

You'll receive new applications, power tips and special reports designed to add more power and flexibility to your IBM/MS DOS-standard system.

And you'll network with other users and discover routines and techniques that can save you time and money.

Subscribe to *PC Magazine* today! Send the coupon to: *PC Magazine* · P.O. Box 2443 · Boulder, Colorado 80322 *or call Toll-Free; 1-800-852-5200 now!*

Yes, send me PC Magazine for:

☐ 22 issues (one year) for $24.97. Save 62%!

☐ 44 issues for only $44.97. Save 65%!

Savings based on annual single-copy price of $64.90.

Mr./Mrs./Ms.______________________________
please print full name

Company______________________________

Address______________________________

City________________State________Zip__________

Add $15 per year for postage outside USA, US currency only.
Please allow up to 60 days for delivery of first issue. Basic annual subscription price is $34.97.

8ZD74

INDEX TO PRODUCTS

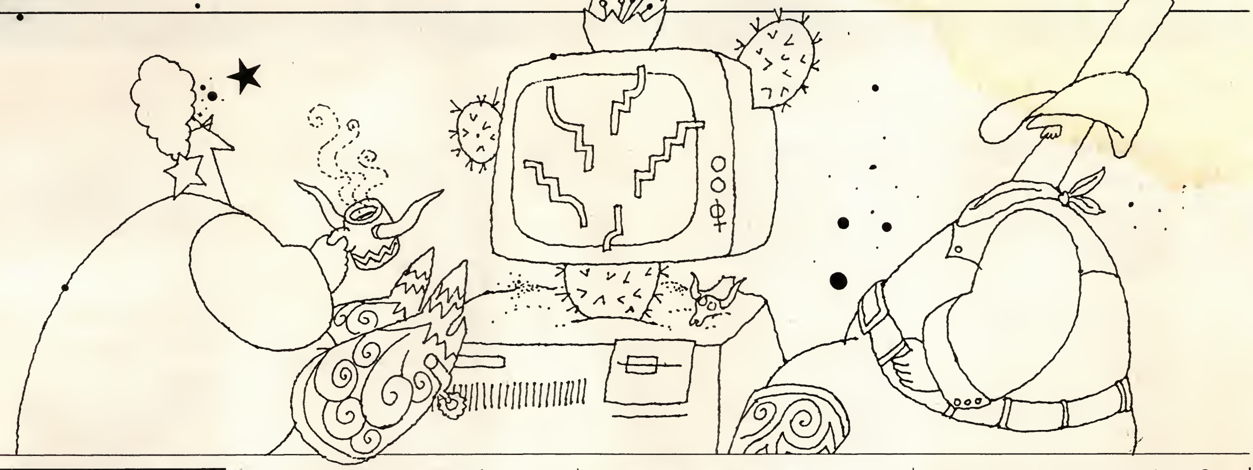


ILLUSTRATION • DAVID POVILAITIS

AUGUST

August 4-6
LISP and Functional Programming
Cambridge, MA
Sponsor: ACM
Contact: Richard P. Gabriel, Lucid, Inc., 707 Laurel Street, Menlo Park, CA 94025

August 5-7
ACM SIGCOMM Futures in Communications Conference
Stowe, VT
Contact: Walter Kosinsky, Norwich University, Northfield, VT 05663; 802/485-5011, ext. 237

August 11-13
Symposium on Principles of Distributed Computing
Calgary, Alberta, Canada
Sponsor: ACM SIGACT-SIGOPS
Contact: ACM, 11 W. 42nd Street, New York, NY 10036

August 18-22
Annual Conference on Computer Graphics and Interactive Techniques
Dallas, TX
Sponsor: SIGGRAPH '86
Contact: Smith, Bucklin, and Andy Associates, 111 E. Wacker Drive, Suite 600, Chicago, IL 60601; 312/644-6610

SEPTEMBER

September 1-5
World Computer Conference
Dublin, Ireland
Sponsor: International Federation for Information Processing
Contact: IFIP Congress '86, 1899 Preston White Drive, Reston, VA 22091; 703/620-8900

September 8-10
NCC-Telecommunications Conference
Philadelphia, PA
Sponsor: AFIPS, ACM, DPMA, IEEE, SCS
Contact: Mike Sherman, AFIPS, 1899 Preston White Drive, Reston, VA 22091; 703/620-8935

September 8-10
1986 ACM Conference on Research and Development in Information Retrieval Systems
Pisa, Italy
Sponsor: ACM SIGIR and AICA-GLIR
Contact: L. Rossi Bernardi, C.N.R. P/zza A. Moro 7, Roma, Italy

September 8-11
National Database and 4th Generation Language Symposium
Hong Kong
Sponsor: Digital Consulting Associates, Inc.
Contact: Digital Consulting Associates, Inc., 6 Windsor Street, Andover, MA 01810; 617/470-3870

September 15-18
Euromicro '86: Symposium on Microprocessing and Microprogramming
Venice, Italy
Contact: Mrs. Chiquita Snippe-Marlisa, p/a TH Twente, Department INF, Room A227, P.O. Box 217, 7500 AE Enschede, The Netherlands; +31/53/338799

September 30-October 2
Northcon '86
Seattle, WA
Sponsor: IEEE
Contact: Northcon '86 Professional Program Committee, Dale Litherland, Director of Education, 8110 Airport Blvd., Los Angeles, CA 90045; 800/421-6816; in California, 800/262-4208

OCTOBER

October 5-10
Compsac '86
Chicago, IL
Contact: Compsac '86, 1730 Massachusetts Avenue NW, Washington, DC 20036-1903; 202/371-0101

October 5-10
International Conference on Computer Design
Atlantic City, NJ
Contact: ICCD '86, 1730 Massachusetts Avenue NW, Washington, DC 20036-1903; 202/371-0101

October 7-9
Introduction to Data Communications
Atlanta, GA
Contact: Deidre Mercer, Department of Continuing Education, Georgia Institute of Technology, Atlanta, GA 30332-0385; 404/894-2547

October 15-17
Scan-Tech '86
San Francisco, CA
Sponsor: Automatic Identification Manufacturers, Inc.
Contact: Automatic Identification Manufacturers, Inc., 1326 Freeport Road, Pittsburgh, PA 15238; 412/963-8588

October 16
Communications Network Procurement
Atlanta, GA
Sponsor: Georgia Institute of Technology
Contact: Deidre Mercer, Department of Continuing Education, Georgia Institute of Technology, Atlanta, GA 30332-0385; 404/894-2547

October 20-24
Using Computer Graphics to Enhance Productivity and Management Effectiveness
Atlanta, GA
Contact: CGA '86 Conference Director, 2033 M Street NW, Suite 333, Washington, DC 20036; 202/775-9556

CALLS FOR PAPERS

August 31, 1986
Computers and Applications
Peking, China
(June 24-26)
Sponsor: IEEE-CS and Chinese Computer Federation
Submit papers to: Oscar N. Garcia, Department of EE and CS, Room T-637, The George Washington University, 801 22nd Street NW, Washington, DC 20052; 202/676-7175

Use these reader service cards to get
F R E E   I N F O R M A T I O N
about the products and services in this issue of TECH JOURNAL

Learning more about a product that's advertised or mentioned in an article in this month's issue is as simple as 1-2-3. And absolutely free.

1 Print or type your name and address on the attached card. Use only one card per person.

2 Circle the numbers on the card that correspond to the numbers at the bottom of the advertisements or articles for which you want more information.

(Key numbers for advertised products also appear in the Advertisers' index.)

3 Simply mail the card, and the literature will be mailed to you free of charge by the manufacturer.

TECH JOURNAL

Are you personally involved in the selection of microcomputers and related products for:

1 Your company or organization?
A☐ Yes B☐ No

2 Your client companies or organizations?
C☐ Yes D☐ No

3 Are you planning to purchase in the next 6 months:
E☐ PC Hardware?
F☐ PC Software?
G☐ PC Peripherals?

101	116	131	146	161	176	191	206	221	236	251	266	281	296	311	326	341	356	371	386
102	117	132	147	162	177	192	207	222	237	252	267	282	297	312	327	342	357	372	387
103	118	133	148	163	178	193	208	223	238	253	268	283	298	313	328	343	358	373	388
104	119	134	149	164	179	194	209	224	239	254	269	284	299	314	329	344	359	374	389
105	120	135	150	165	180	195	210	225	240	255	270	285	300	315	330	345	360	375	390
106	121	136	151	166	181	196	211	226	241	256	271	286	301	316	331	346	361	376	391
107	122	137	152	167	182	197	212	227	242	257	272	287	302	317	332	347	362	377	392
108	123	138	153	168	183	198	213	228	243	258	273	288	303	318	333	348	363	378	393
109	124	139	154	169	184	199	214	229	244	259	274	289	304	319	334	349	364	379	394
110	125	140	155	170	185	200	215	230	245	260	275	290	305	320	335	350	365	380	395
111	126	141	156	171	186	201	216	231	246	261	276	291	306	321	336	351	366	381	396
112	127	142	157	172	187	202	217	232	247	262	277	292	307	322	337	352	367	382	397
113	128	143	158	173	188	203	218	233	248	263	278	293	308	323	338	353	368	383	398
114	129	144	159	174	189	204	219	234	249	264	279	294	309	324	339	354	369	384	399
115	130	145	160	175	190	205	220	235	250	265	280	295	310	325	340	355	370	385	400

Please print clearly—Use only one card per person. Void after November 30, 1986

Name__________ Phone (____)__________

Title__________

Company__________

Address__________ Apt.____

City__________ State______ Zip______

(Zip code must be included to insure delivery.)

4☐ **Please send me 1 year (12 issues) of *PC Tech Journal* for $23.70 and bill me. I'll save 50% off the single copy price!**

TJ8862

TECH JOURNAL

Are you personally involved in the selection of microcomputers and related products for:

1 Your company or organization?
A☐ Yes B☐ No

2 Your client companies or organizations?
C☐ Yes D☐ No

3 Are you planning to purchase in the next 6 months:
E☐ PC Hardware?
F☐ PC Software?
G☐ PC Peripherals?

101	116	131	146	161	176	191	206	221	236	251	266	281	296	311	326	341	356	371	386
102	117	132	147	162	177	192	207	222	237	252	267	282	297	312	327	342	357	372	387
103	118	133	148	163	178	193	208	223	238	253	268	283	298	313	328	343	358	373	388
104	119	134	149	164	179	194	209	224	239	254	269	284	299	314	329	344	359	374	389
105	120	135	150	165	180	195	210	225	240	255	270	285	300	315	330	345	360	375	390
106	121	136	151	166	181	196	211	226	241	256	271	286	301	316	331	346	361	376	391
107	122	137	152	167	182	197	212	227	242	257	272	287	302	317	332	347	362	377	392
108	123	138	153	168	183	198	213	228	243	258	273	288	303	318	333	348	363	378	393
109	124	139	154	169	184	199	214	229	244	259	274	289	304	319	334	349	364	379	394
110	125	140	155	170	185	200	215	230	245	260	275	290	305	320	335	350	365	380	395
111	126	141	156	171	186	201	216	231	246	261	276	291	306	321	336	351	366	381	396
112	127	142	157	172	187	202	217	232	247	262	277	292	307	322	337	352	367	382	397
113	128	143	158	173	188	203	218	233	248	263	278	293	308	323	338	353	368	383	398
114	129	144	159	174	189	204	219	234	249	264	279	294	309	324	339	354	369	384	399
115	130	145	160	175	190	205	220	235	250	265	280	295	310	325	340	355	370	385	400

Please print clearly—Use only one card per person. Void after November 30, 1986

Name__________

Title__________ Phone (____)__________

Company__________

Address__________ Apt.____

City__________ State______ Zip______

(Zip code must be included to insure delivery.)

4☐ **Please send me 1 year (12 issues) of *PC Tech Journal* for $23.70 and bill me. I'll save 50% off the single copy price!**

TJ8861

FREE INFORMATION

Follow the instructions
on the reverse side of
this card to
receive advertisers'
product information.
FREE.

SUBSCRIBE NOW!

Now's an ideal time to consider having us start you as a PC Tech Journal subscriber.
12 issues cost you only \$23.70...a savings of 50% off the single copy price. Just check the box at the bottom of the reply card.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 27346 PHILADELPHIA, PA

Postage will be paid by addressee

TECH JOURNAL
P.O. Box 40086
Philadelphia, PA 19106-9931

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 27346 PHILADELPHIA, PA

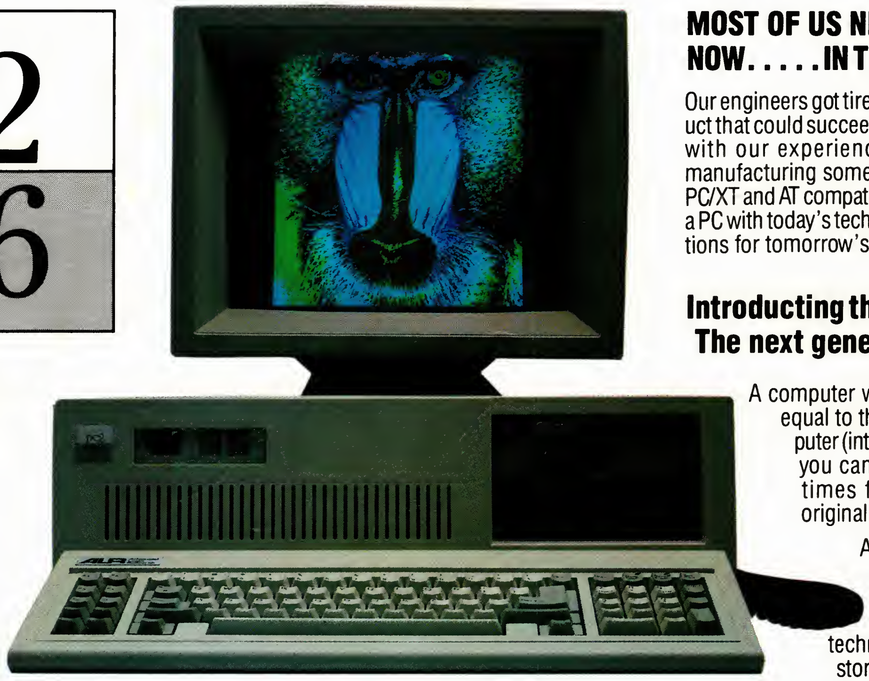
Postage will be paid by addressee

TECH JOURNAL
P.O. Box 40086
Philadelphia, PA 19106-9931

Sometimes Waiting For Large Corporations To Introduce New Technology Is Like Watching Apes Evolve

pc2
286

PC2/286
$1545
PC/XT Size Shown with Optional ALR EGA Graphics



MOST OF US NEED SOLUTIONS NOW. . . . . IN THIS MILLENIUM!

Our engineers got tired of waiting for a product that could succeed the legendary PC. So with our experience in designing and manufacturing some of the most powerful PC/XT and AT compatibles, we've engineered a PC with today's technology to provide solutions for tomorrow's challenges.

Introducting the ALR PC2/286. The next generation.

A computer with processing power equal to the 8MHz IBM AT computer (introduced April 1986), so you can solve problems 500 times faster than with the original PC.

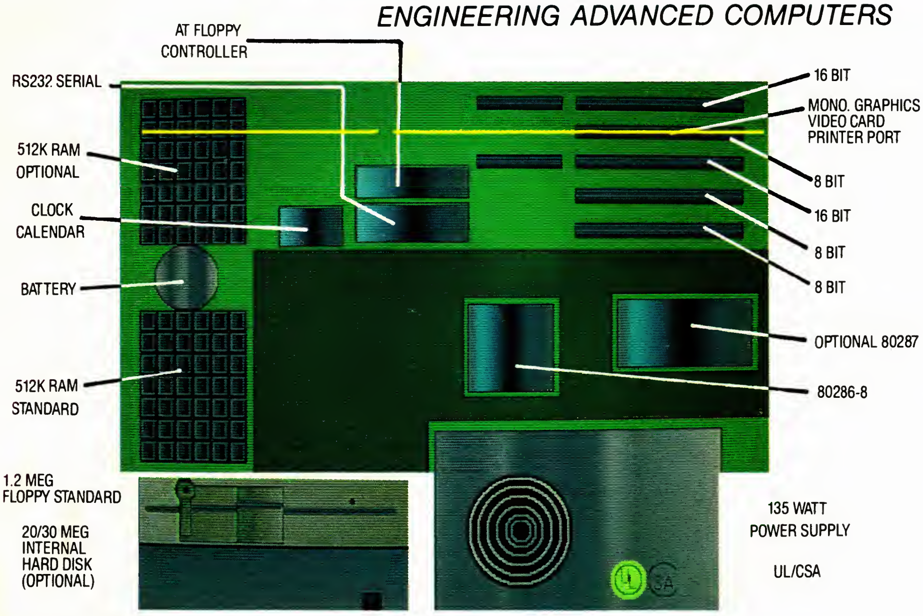
A computer with more storage capacity. The PC2/286 integrates modern floppy technology allowing you to store 3 times more information than you could before.

Evolutionary Winchester hard disk technology further enhances the power of the PC2/286.

The PC2/286 incorporates all the best hardware developments of yesterday while maintaining the vital link of total software compatibility through Phoenix Bios.

The economy of modern technology allows the PC2/286 to be priced lower than the original PC and most other primitive PC compatibles.

ENGINEERING ADVANCED COMPUTERS



ADD UP TO 2 HALF-HEIGHT FLOPPIES
2 HALF-HEIGHT OR 1 FULL HEIGHT HARD DISK

The Result is a $1545 Complete System.

If you are tired of waiting for a maybe product, or you don't want to buy yesterday's technolgy, then treat yourself to the performance and features of the PC2/286.

Call and talk to our dealers or sales engineers for the product that says tomorrow.

ALR IN CANADA 416-229-6477
DEALER INFORMATION CONTACT: MR. KIRKEY or MR. ANEIL

Advanced Logic Research, Inc.
10 Chrysler, Irvine, California 92718 - (714) 581-6770
FAX: (714) 581-9240 - TELEX: 5106014525, Answer back Advanced Logic

CIRCLE NO. 116 ON READER SERVICE CARD